

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ С.А. Иванов

«__» _____ 2022 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«__» _____ 2022 г.

Программно-аппаратный комплекс для организации соревнований и тренировок
по современному пятиборью

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К МАГИСТЕРСКОЙ ДИССЕРТАЦИИ
ЮУРГУ-090401.2022.210 ПЗ ВКР

Руководитель работы,

к.п.н., доцент каф. ЭВМ

_____ Ю.Г. Плаксина

«__» _____ 2022 г.

Автор работы,

студент группы КЭ-222

_____ М.А. Петрова

«__» _____ 2022г.

Нормоконтролёр,

ст. преп. каф. ЭВМ

_____ С.В. Сяськов

«__» _____ 2022г.

Челябинск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2022 г.

ЗАДАНИЕ
на магистерскую диссертацию
группы КЭ-222
Петровой Марии Александровны,
обучающейся по направлению
09.04.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Программно-аппаратный комплекс для организации соревнований и тренировок по современному пятиборью», утверждена приказом по университету от 25 апреля 2022 г. №697-13/12.

2. **Срок сдачи студентом законченной работы:** 1 июня 2022 г.

3. **Исходные данные к работе**

Перечень функционала разрабатываемого приложения:

- 1) возможность создания и редактирования официальных и тренировочных соревнований;
- 2) фиксация результатов по нескольким видам многоборья;
- 3) интерпретация результатов по всем видам многоборья в балльно-рейтинговую систему;

- 4) оформление итоговых и промежуточных протоколов по результатам соревнований;
- 5) возможность регистрации и авторизации пользователей; предоставление информации различным пользователям через «Личный кабинет» (спортсменам, тренерам, организаторам соревнований);
- 6) составление календаря тренировок/соревнований спортсменам через личный кабинет; отслеживание динамики изменений личных параметров спортсмена и результатов по всем соревнованиям, в которых он участвовал;

4. Перечень подлежащих разработке вопросов:

- 1) анализ предметной области;
- 2) анализ существующих аналогов, выявление их преимуществ и недостатков;
- 3) формулирование требований к разрабатываемому комплексу;
- 4) разработка структуры программно-аппаратного комплекса;
- 5) проектирование аппаратной части комплекса:
 - разработка структурной схемы модуля сопряжения;
 - формулирование требований и подбор элементов;
 - разработка принципиальной схемы модуля сопряжения.
- 6) разработка программной части комплекса:
 - выбор среды разработки и средств реализации;
 - разработка архитектуры веб-приложения;
 - проектирование структуры базы данных;
 - реализация функционала веб-приложения.

5. **Дата выдачи задания:** 1 декабря 2021 г.

Руководитель работы _____ / Ю.Г. Плаксина /

Студент _____ / М.А. Петрова /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Анализ предметной области	10.01.2022	
Анализ существующих аналогов, выявление их преимуществ и недостатков	25.01.2022	
Формулирование требований к разрабатываемому комплексу	15.02.2022	
Разработка структуры программно-аппаратного комплекса	01.03.2022	
Проектирование аппаратной части комплекса	25.03.2022	
Разработка программной части комплекса	15.04.2022	
Тестирование работы программно-аппаратного комплекса, отладка и эксперименты	15.05.2022	
Компоновка текста работы и сдача на нормоконтроль	24.05.2022	
Подготовка презентации и доклада	30.05.2022	

Руководитель работы _____ / Ю.Г. Плаксина /

Студентка _____ / М.А. Петрова /

АННОТАЦИЯ

М.А. Петрова. Программно-аппаратный комплекс для организации соревнований и тренировок по современному пятиборью. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2022, 104 с., 52 ил., 14 табл., библиогр. список – 23 наим., 1 прил.

В данной выпускной квалификационной работе выполнена разработка программно-аппаратного комплекса для организации соревнований и тренировок по современному пятиборью.

В ходе работы была изучена предметная область, выполнен обзор существующих аналогов, выявлены их преимущества и недостатки. На основании этого были сформулированы требования к разрабатываемому комплексу, его программной и аппаратной части. При выполнении программной части был произведен анализ современных технологий создания веб-приложений, сделан выбор технологий на основе языков программирования JavaScript и PHP – TailwindCSS с использованием HTML- и CSS-шаблонов.

При выполнении аппаратной части была разработана структурная и принципиальная схемы модулей сопряжения стрелковой мишени и фехтовального оборудования.

Пояснительная записка состоит из введения, пяти основных частей, заключения, библиографического списка и приложений.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	10
1.1. ОБЗОР АНАЛОГОВ	10
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ	20
1.2.1. Выбор СУБД	20
1.2.2. Выбор фреймворка для веб-приложения	24
1.2.3. Выбор языков программирования	28
1.3. ВЫВОД	29
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	30
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	30
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	32
2.2.1. Общие требования	32
2.2.2. Требования к аппаратной части комплекса	32
2.2.3. Требования к надежности	33
2.2.4. Требования к документации	33
2.2.5. Требования к безопасности	33
3. ПРОЕКТИРОВАНИЕ	34
3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ	34
3.1.1. Архитектура аппаратной части комплекса	34
3.1.2. Архитектура программной части комплекса	47
3.2. ОПИСАНИЕ ДАННЫХ	51
4. РЕАЛИЗАЦИЯ	56
5. ТЕСТИРОВАНИЕ	73
ЗАКЛЮЧЕНИЕ	82
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	83
ПРИЛОЖЕНИЕ А	86

ВВЕДЕНИЕ

Современное пятиборье включает в себя 5 дисциплин: плавание, фехтование, конкур (верховая езда) и комбайн (вид, который представляет из себя комбинацию из стрельбы и бега). Данный вид спорта появился во второй половине 19 века в Швеции, но его предшественником по праву считается античное пятиборье, появившееся в 708 году до н.э, которое включало в себя бег, метание копья, метание диска, прыжок и борьбу.

Раньше современное пятиборье называли офицерским, поскольку вплоть до 1948 г. участие в нем могли принимать лишь спортсмены-офицеры.

Комплекс офицерского пятиборья разработал французский спортивный и общественный деятель Пьер де Кубертен, желая именно таким способом проверять «моральные качества» и «физические возможности и навыки» спортсмена для определения «идеального, разностороннего атлета» [1].

Как правило, официальные индивидуальные соревнования организованы так, чтобы все пять видов многоборья проходили в один день, в одном месте, в шаговой доступности друг от друга. Командный зачет в современном пятиборье включает начисленные очки индивидуального первенства. Тренеры обязаны информировать организаторов соревнований, какие спортсмены входят в состав команды.

АКТУАЛЬНОСТЬ РАБОТЫ

Соревнования по современному пятиборью были включены в программу Олимпийских игр в 1912 году в Стокгольме. Соревнования у женщин появились позднее, только в 2000 году в Сиднее.

Современное пятиборье широко развивается в РФ. В настоящий момент насчитывается более 30 Федерации Регионального уровня, практически каждая из которых проводит Всероссийские и Международные соревнования.

Промежуточные результаты по каждому из видов современного пятиборья на одном соревновании фиксируются различными судьями с аппаратуры, после чего передаются секретарю соревнований в «ручном» режиме для внесения в протокол, что часто приводит к возникновению ошибок.

Перевод результатов в балльную систему по каждому виду многоборья определяется с использованием не стандартизированных программ. Также нет возможности оценить выполнение каждого компонента по отдельным видам, которые не нужны для выявления лидеров и присвоения дополнительных баллов на соревновании, но являются необходимыми при качественной подготовке спортсмена и получения наивысшего результата в дальнейшей работе. Такими компонентами являются, например: качество и скорострельность при стрельбе, бег за каждые 800 м дистанции (из четырех возможных отрезков) и т.п.

При проведении соревнований по современному пятиборью возникают следующие проблемы:

1. Высокая трудоемкость регистрации результатов.
2. Большие временные затраты на обработку результатов.
3. Возникновение ошибок при регистрации и передаче данных.

На данный момент на территории РФ не разработан единый программно-аппаратный комплекс, применяемый в соревнованиях и тренировках по современному пятиборью, при этом актуальность в его создании подтверждается заинтересованностью Федерацией современного пятиборья Челябинской области и МБУ «Спортивная школа Конноспортивный комплекс Рифей им. П.М.Латышева» г. Челябинска (в состав данной школы входит спортивная школа по современному пятиборью) [2].

ЦЕЛИ И ЗАДАЧИ РАБОТЫ

Целью представленной выпускной квалификационной работы является разработка программно-аппаратного комплекса, предназначенного для организации соревнований и тренировок по современному пятиборью.

Поставленная цель будет достигнута путем решения следующих задач:

1. Анализ предметной области.
2. Анализ существующих аналогов, выявление их преимуществ и недостатков.
3. Формулирование требований к разрабатываемому комплексу.
4. Разработка структуры программно-аппаратного комплекса.
5. Проектирование аппаратной части комплекса:
 - разработка структурной схемы модуля сопряжения;
 - формулирование требований и подбор элементов;
 - разработка принципиальной схемы модуля сопряжения.
6. Разработка программной части комплекса:
 - выбор среды разработки и средств реализации;
 - разработка архитектуры веб-приложения;
 - проектирование базы данных;
 - реализация функционала веб-приложения.
7. Произвести тестирование работы программно-аппаратного комплекса.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. ОБЗОР АНАЛОГОВ

1.1.1 Стрелковое оборудование

В настоящее время на рынке существует ряд технических решений для проведения соревнований по всем пяти видам современного пятиборья. В рамках данной работы будут рассмотрены аппаратные решения только для одного/двух из видов многоборья (стрельбы и фехтования).

При проведении стрелковой части многоборья используется мишень и лазерный пистолет, каждый из которых имеет сертификат соответствия UIPM (Union Internationale de Pentathlon Moderne, Международный Союз современного пятиборья).

В соответствии с требованиями UIPM в соревновательном процессе может быть использовано только сертифицированное оборудование, с которым можно ознакомиться на официальном сайте [12].

На данный момент существует два основных вида стрелковых мишеней: «Hit&Miss» (рисунок 1.5) и «Laser Precision» (рисунок 1.1). Данные мишени могут регистрировать факт выстрела и отображать результат попадания на табло, однако они не имеют возможности передавать данные беспроводной способом. Следовательно, возникает необходимость в разработке средства, способного принимать данные с мишени, интерпретировать их и передавать посредством Wi-Fi в Облако.

Рассмотрим принцип работы стрелковой мишени на примере точной мишени S-BOX (рисунок 1.1), выпускаемой фирмой PENTASHOT (Чехия) с одноименным программным продуктом [3]. Технические характеристики стрелковой мишени S-BOX представлены в таблице 1.1.

Мишень может быть использована в двух режимах: тренировочном и соревновательном (рисунок 1.3). Она способна анализировать точки вне попадания в зону попадания или попадания во внешнюю зону мишени. Зона попадания обозначается загоранием зеленого светодиода и виртуально отображается на экране. Попадание во внешнюю зону мишени обозначается включением / зажиганием красного светодиода и также виртуально фиксируется на экране. Допустимый интервал времени стрельбы составляет 50 с после первого выстрела.



Рисунок 1.1 – Стрелковая мишень S-BOX

Таблица 1.1 – Технические характеристики стрелковой мишени S-BOX

Название параметра	Величина
Размер мишени, мм	205 · 190 · 250
Размеры зоны попадания, мм	59,5
Вес, кг	1,65
Расстояние до стрельбы, м	10
Тип лазерного сигнала, мс	Лазерный сигнал UIPM 15,6 & 25,2
Внешние сигнальные огни	Наружные светильники марки PENTASHOT
Формат выходных данных	UIPM Open target protocol

Левый светодиод из пяти возможных по истечению 40 с начнет медленно мигать, по истечению 45 с начнет мигать быстро. Все светодиоды по истечению 50 с начнут на 5 с мигать. Также все светодиоды начинают мигать, если спортсмен заработает 5 очков за попадание внутрь зоны попадания. На данной мишени не предусмотрено отображение попаданий, кроме как через специальное приложение, но возможно подключить специальные сигнальные лампы для визуализации стрельбы (рисунок 1.2).



Рисунок 1.2 – Сигнальные лампы для стрелковой мишени

При первом подключении мишени при помощи USB-кабеля к PC необходимо установить приложение.

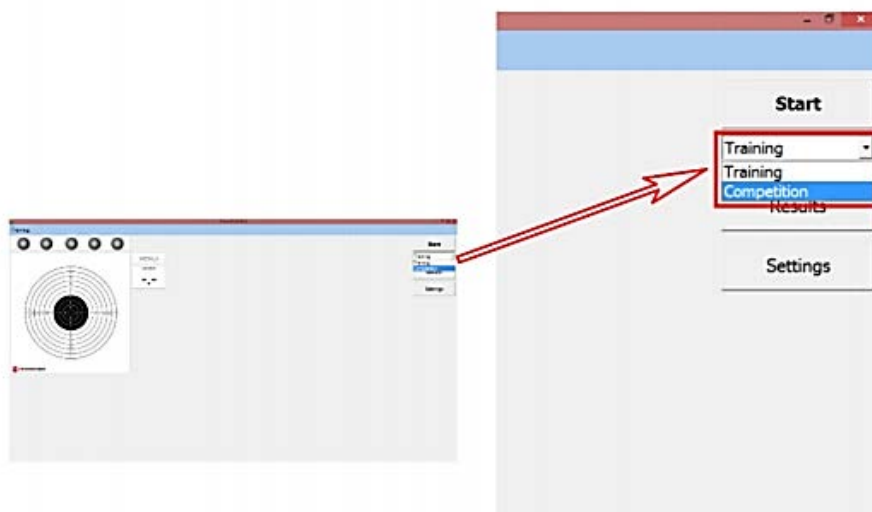


Рисунок 1.3 – Тренировочный режим

В соревновательном режиме результаты можно выводить на экран по одному раунду или по несколько (рисунок 1.4) [3].

SBox2 Competition Result

Event parameters: 23. 10. 2015 10.01.27: Competition on Range 1, target number 1

	Score	AHP	Time
Round 1	54,5	9,0	5,96s
Round 2	54,3	9,1	5,12s
Round 3	66,7	8,9	13,03s
Round 4	48,1	9,6	3,26s

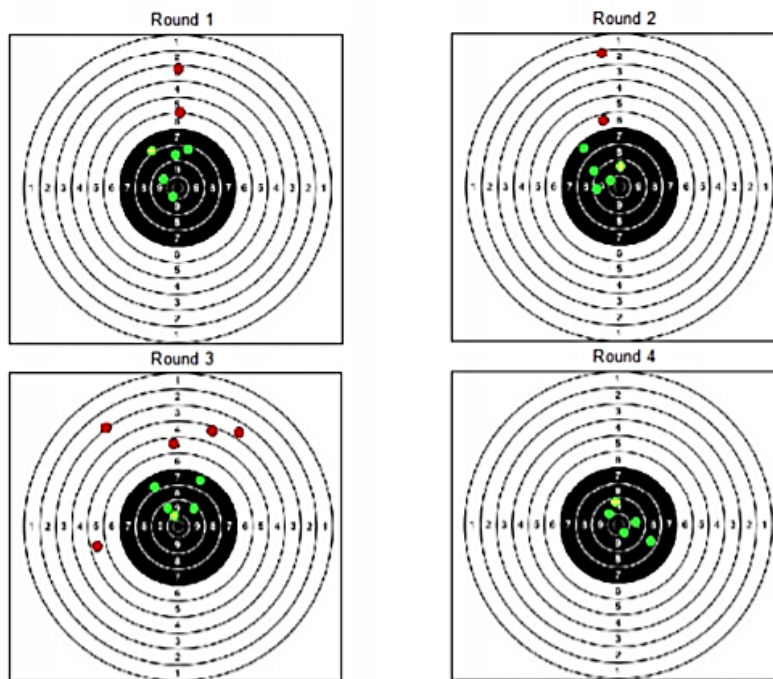


Рисунок 1.4 – Соревновательный режим

До 1.01.2009 г. стрельба оценивалась отдельно от бега и давала самостоятельные очки, поэтому для тренеров и зрителей соревнований было важно знать стрелковое время каждого спортсмена. После объединения бега со стрельбой (далее комбайн) необходимость в регистрации «чистого» стрелкового времени отпала, так как оно стало частью комбинированного времени.

В связи с этим актуальность использования программно-аппаратного комплекса PENTASHOT для тренировочного процесса остается, а для соревновательного есть необходимость использовать только его аппаратную часть, состоящую из электронной мишени. Для наглядности все технические характеристики были вынесены в таблицу 1.2.

Таблица 1.2 – Сравнение технических характеристик мишени S-BOX и разрабатываемой аппаратной части комплекса

Технические характеристики	Мишень S BOX	Комплекс
Способность анализировать выстрелы вне зоны попадания или попадания во внешнюю зону мишени	+	+
Обозначение зон «попадания»/«непопадания» на табло и экране компьютера зелёным/красным светодиодом	+/+	+/-
Контроль допустимого интервала времени стрельбы от первого выстрела	+	+
Длина кабеля подключения через USB интерфейс, м	5	0
Наличие промышленного интерфейса для подключения до 15 устройств	-	+
Совместимость с излучателями, имеющими кодовую сертификацию UIPM	+	+

Продолжение таблицы 1.2

Технические характеристики	Мишень S BOX	Комплекс
Сохранение результатов в единой базе данных	-	+
Статистическая обработка результатов соревновательной деятельности спортсменов	-	+
Предоставление информации различным пользователям (спортсменам, тренерам, организаторам соревнований)	-	+
Применимость к виду спорта	стрельба	пятиборье
Наличие полуавтоматической системы хронометража по плаванию	-	+
Наличие средств электрофиксации уколов в фехтовании	-	+
Единая программно-аппаратная среда ведения документации для современного биатла, триатла, тетратла и пятиборья	-	+
Соответствие требованиям безопасности передачи и хранения данных	-	+

Мишень Hit&Miss (рисунок 1.5), выпускаемая той же фирмой PENTASHOT (Чехия), что и модель S-BOX.

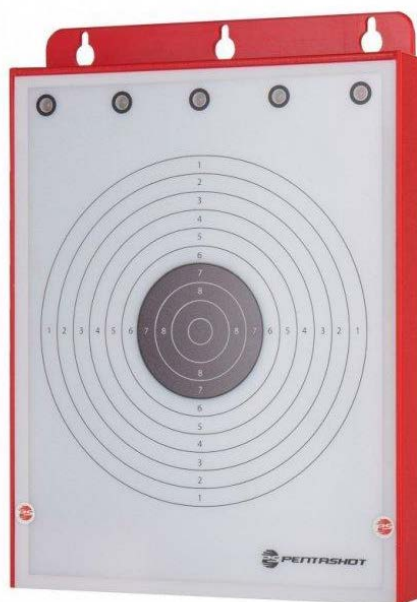


Рисунок 1.5 – Стрелковая мишень Hit&Miss

Основной отличительной ее чертой является возможность использования только в соревновательном режиме, т.к. она способна определить только попадание/непопадание выстрела и отображать результат с помощью светодиода (красный – «промах», зеленый – «попадание»). Технические характеристики мишени представлены в таблице 1.3.

Таблица 1.3 – Технические характеристики стрелковой мишени Hit&Miss

Название параметра	Величина
Размер мишени, мм	270 · 190 · 32
Размеры зоны попадания, мм	59,9
Вес, кг	0,7
Расстояние до стрельбы, м	10
Тип лазерного сигнала, мс	Лазерный сигнал UIPM 15,6 & 25,2
Внешние сигнальные огни	NAVISTREET
Формат выходных данных	UIPM Open target protocol

Данная модель реализована в виде конструктива без возможности передачи результатов на РС.

1.1.2 Программное обеспечение по современному пятиборью

В настоящее время разработано программное обеспечение SportGeek для проведения соревнований по современному биатлону, триатлону, тетратлону и пятиборью, разрабатываемое в Великобритании с 2014 г [4].

Интерфейс программного обеспечения SportGeek представлен на рисунке 1.6.

Competitor	Fence	Swim	Ride	Combined Event
6 - Susannah Boddie University of Cambridge Total Score: 1094	13 26 0	2:53.59 203 points	68secs, 0 J.Pens, 0 O.Pens 300 points	15:19.00 381 points
7 - Kerenza Bryson University of Plymouth Total Score: 1251	23Vs 3Ds 290 points	2:24.82 260 points	64secs, 0 J.Pens, 0 O.Pens 300 points	13:19.00 501 points
8 - Amy Bull Oxford University Total Score: 947	11Vs 15Ds 194 points	2:56.22 198 points	92secs, 17 J.Pens, 0 O.Pens 267 points	16:52.00 288 points
9 - Charlotte Coles University of Cambridge Total Score: 363	7Vs 19Ds 162 points	2:55.88 198 points	0secs, 0 J.Pens, 0 O.Pens - points	18:17.00 203 points
10 - Clemmie Cooper Loughborough University Total Score: 1157	16Vs 10Ds 234 points	2:42.47 225 points	62secs, 7 J.Pens, 0 O.Pens 293 points	14:55.00 405 points
11 - Abbie Currington University of Cambridge Total Score: 654	7Vs 19Ds 162 points	2:38.19 234 points	0secs, 0 J.Pens, 0 O.Pens - points	17:22.00 258 points
12 - Samantha Farnfield University of Birmingham Total Score: 785	10Vs 16Ds 186 points	2:17.25 276 points	0secs, 0 J.Pens, 0 O.Pens - points	16:17.00 323 points
13 - Anna Gibbons University of Cambridge Total Score: 877	15Vs 11Ds 226 points	2:28.56 253 points	0secs, 0 J.Pens, 300 O.Pens [Terminated]	16:02.00 338 points
14 - Emma Godden Oxford University Total Score: 501	5Vs 21Ds 146 points	2:45.09 220 points	0secs, 0 J.Pens, 300 O.Pens [Terminated]	19:25.00 135 points
15 - Stephanie Gover University of Cambridge Total Score: -	0Vs 0Ds DNF points [DNF]	DNF	0secs, 0 J.Pens, 0 O.Pens - points	DNF
16 - Olivia Green University of Bath Total Score: 1277	22Vs 4Ds 282 points	2:17.47 275 points	88secs, 37 J.Pens, 0 O.Pens 251 points	13:51.00 469 points

Рисунок 1.6 – Пример работы программного обеспечения SportGeek

К недостаткам данного программного обеспечения можно отнести:

- отсутствие единой программно-аппаратной среды ведения документации;

- нет поддержки программного обеспечения на русском языке;
- нет возможности использования в тренировочном процессе;
- программное обеспечение разрабатывается за счет грантовой поддержке с 2014 г. и до сих пор находится на стадии апробации, что не дает возможности использовать его официально на соревнованиях.

1.1.3 Фехтовальное оборудование

При проведении фехтовальной части многоборья используется следующее оборудование: токопроводящие фехтовальные дорожки (металлические или прорезиненные), электрофиксатор уколов, катушки-сматыватели с подводящими подводящие шнурами (кабелями) на 20 м.

На данный момент наиболее часто используется многофункциональный электрофиксатор уколов по фехтованию фирмы Favero FA-07 (производство-Италия) (Рисунок 1.7).

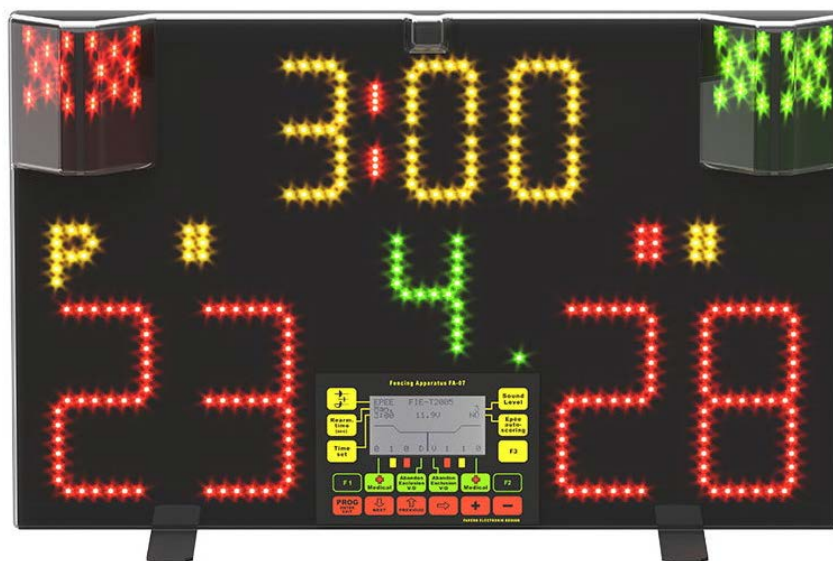


Рисунок 1.7 –Электрофиксатор уколов по фехтованию Favero FA-07

Данный электрофиксатор уколов подтвержден Международной Федерацией фехтования. Он позволяет производить отсчет временного интервала и фиксирование попаданий (уколов). Также имеет 3 уровня громкости звуковых сигналов. В набор с аппаратурой входит пульт дальностного управления. С помощью него можно выбрать вид оружия, настраивать хронометраж и счет. Питание пульта обеспечивается от батареек [22].

В таблице 1.4 приведены основные технические характеристики данного электрофиксатора уколов.

Таблица 1.4 – Технические характеристики электрофиксатора уколов по фехтованию Favero FA-07

Материал	ударостойкий пластик, поликарбонат
Параметры	60 см x 40 см x 13,5 см
Цвет	черный
Вес	8,1 кг
Сертификация	FIE (Fédération Internationale d'Escrime)
Что входит в набор с ним	подзарядка, защитный кейс, пульт
Виды оружия	сабля, шпага и рапира
Страна изготовления	Италия

Данный электрофиксатор реализован в виде конструктива без возможности передачи результатов на РС.

1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

1.2.1. Выбор СУБД

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. Каждая база данных имеет определённую модель (реляционная, сетевая, документно-ориентированная и др.), которая обеспечивает удобный доступ к данным. База данных обычно управляется системой управления базами данных (СУБД).

Система управления базами данных (СУБД) — специализированная программа (чаще комплекс программ), предназначенная для организации и ведения БД.

К критериям выбора СУБД можно отнести: масштабируемость, возможность веб-разработки, безопасность, качество и полнота документации, распространённость СУБД, гибкость и мощность [5].

Далее рассмотрим наиболее распространённые СУБД, которые могут быть использованы для реализации программной части комплекса.

СУБД **SQLite** [6] имеет отличительную особенность, которая позволяет встраивать ее в приложение. Это значит, что большинство СУБД являются самостоятельными приложениями, взаимодействие с которыми организовано по принципу клиент-сервер. Программа-клиент посылает запрос на языке SQL, СУБД, которая в том числе может находиться на удалённом компьютере, возвращает результат запроса.

В свою очередь SQLite является написанной на языке C библиотекой, которую динамически или статически подключают к программе.

Благодаря технологиям обслуживающих библиотек SQLite обладает такими качествами, как быстрота и мощность. Рассмотрим более подробно преимущества и недостатки данной СУБД.

Преимущества SQLite:

- простота файловой структуры – вся БД состоит из одного файла, из-за чего ее легко переносить на другие машины;
- использование стандартов – поддержание основных особенностей SQL, кроме RIGHT OUTER JOIN или FOR EACH STATEMENT;
- подходит для разработки и тестирования – так как SQLite состоит всего лишь из одного файла и библиотеки, написанной на С, то не составит труда масштабировать приложение в процессе разработки.

Недостатки SQLite:

- отсутствие системы пользователей. Крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях;
- отсутствие возможности увеличения производительности;
- SQLite стоит использовать в том случае, если важна возможность переноса приложения, масштабируемость и необходимо напрямую обращаться к диску.

Данная СУБД не подойдет для работы, если планируется разработать многопользовательское приложение, в котором присутствует разграничение прав пользователей и идет работа с большими объемами данных.

СУБД **PostgreSQL** [6] свободно распространяема и максимально соответствует стандартам SQL. В PostgreSQL стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий.

СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций (атомарность, последовательность, изоляционность, прочность). Данная СУБД имеет высокий уровень производительности. Параллельность достигается не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом. PostgreSQL очень легко расширять своими процедурами, которые называются - хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций.

Преимущества PostgreSQL:

- бесплатное программное обеспечение (ПО) с открытым исходным кодом;
- объемная справочная система;
- большое количество встроенных функций и дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;
- предусмотрена возможность расширения функционала за счет сохранения своих процедур;
- объектно-ориентированная СУБД с поддержкой наследования.

Недостатки PostgreSQL:

- невысокая производительность при простых операциях чтения, что замедляет сервер и делает медленнее относительно конкурентов, таких как MySQL;
- не так популярна, как другие СУБД;
- сложности в поиске хостинга с поддержкой данной СУБД.

Если при выборе СУБД главными критериями являются целостность данных и использование сложных пользовательских процедур, то стоит остановить свой выбор на PostgreSQL.

Если такими же важными критериями являются скорость и репликация, то стоит подобрать другую СУБД.

СУБД **MySQL** [6] является одной из самых популярных и самых распространенных серверных. Данная СУБД отлично подходит для проектирования каких-либо веб-приложений и сайтов. MySQL имеет большое количество официальной документации, что упрощает процесс обучения работы с данной программой.

Преимущества MySQL:

- простой процесс установки и наличие дополнительных приложений (например, GUI) позволяет без проблем работать с MySQL;
- поддержка большинства функционала SQL;
- большое количество функций, обеспечивающих безопасность, которые поддерживаются по умолчанию;
- MySQL легко работает с большими объемами данных и легко масштабируется;
- имеет высокую производительность за счет упрощения определенных стандартов;
- большое количество официальной документации;
- бесплатное распространение.

Таким образом, MySQL удовлетворяет требованиям, предъявленным к разрабатываемому комплексу, так как ключевыми критериями выбора СУБД являются высокий уровень безопасности, скорость, масштабируемость, работа с распределенными операциями и работа с веб-приложениями.

1.2.2. Выбор фреймворка для веб-приложения

Фреймворк - программное обеспечение, которое позволяет упростить процесс разработки и сборки самых различных модулей (частей) одного программного проекта.

У фреймворков есть две основные функции: работа на серверной стороне (backend) и работа на клиентской стороне (frontend).

Frontend фреймворк связан с внешней частью приложения. Backend связан с внутренним устройством приложения.

На данный момент существует большое количество разнообразных фреймворков для создания веб-приложений. Рассмотрим некоторые из них.

React JS [7] — это библиотека JavaScript, исходный код которой был открыт Facebook в 2013 году. Данный фреймворк подходит для разработки больших веб-приложений, где будут постоянно меняться данные.

Преимущества React:

- простой синтаксис;
- высокий уровень гибкости;
- виртуальная DOM (document object model), которая позволяет упорядочивать документы форматов HTML, XHTML или XML в дерево, которое используется веб-браузером для анализа различных элементов веб-приложения;
- JavaScript-библиотека с открытым исходным кодом, которая получает множество ежедневных обновлений и улучшений в соответствии с отзывами разработчиков по всему миру;
- простая миграция между версиями.

Недостатки React:

- недостаточный объем официальной документации, из-за чего могут возникнуть сложности в изучении данного фреймворка;
- в фреймворке неполное MVC, из-за чего код из React не получится перенести на прочие платформы или библиотеки.

Angular [7] является платформой с открытым исходным кодом, которую можно применять для создания как одностраничных веб-приложений, так и приложений для мобильного телефона. Поддерживает установку зависимостей, комплексную обработку и декларативные шаблоны.

Angular является новой платформой, в основе которой лежит Angular.JS.

Достоинства Angular:

- высокая производительность;
- подробная документация, которая позволяет получить всю необходимую информацию;
- представлена возможность работать отдельно в одном разделе программы, используя имеющиеся данные.

Недостатки Angular:

- в основе сложный язык программирования;
- ошибки во время миграции, которые могут возникнуть при переходе между версиями.

Vue.js [7] – это фреймворк JavaScript, запущенный в 2013 году, который идеально подходит для создания адаптируемых пользовательских интерфейсов и сложных одностраничных приложений.

Достоинства Vue.js

- подробная документация, которая позволяет быстрее изучить данный фреймворк;

- адаптивность □ схожесть с Angular и React с точки зрения дизайна и архитектуры позволяет быстро перейти к Vue.js от других фреймворков;
- интеграция;
- масштабируемость;
- небольшой размер (около 20 Кб), что позволяет обеспечить более высокую скорость и производительность по сравнению с другими фреймворками.

Недостатки Vue.js:

- низкая популярность относительно других фреймворков, таких как React и Angular;
- отсутствие поддержки крупных проектов.

Node.js [9] – это кроссплатформенная среда выполнения JavaScript с открытым исходным кодом, выполняющая код JavaScript вне вашего браузера.

Преимущества Node.js:

- высокая производительность и масштабируемость, достигнуты за счет хорошо разработанных библиотек на основе JavaScript Google Chrome V8;
- возможность использования Node.js как на серверной стороне (backend), так и на клиентской (frontend);
- Node.js имеет асинхронные библиотеки.

Недостатки Node.js:

- не подходит для серверов, требующих высокой производительности процесса, таких как генерирование графики, обработка изображений и сортировка;
- недостаточно полная документация. Не содержит основные библиотеки и инструменты, из-за чего приходится использовать альтернативные варианты.

Tailwind CSS [23] — это настраиваемый низкоуровневый CSS-фреймворк с широкими возможностями настройки, который предоставляет вам все строительные блоки, необходимые для создания дизайна на заказ без каких-либо раздражающих, продуманных стилей, с которыми вам придется бороться, чтобы переопределить.

Особенности Tailwind CSS

- Каждая утилита Tailwind поставляется с адаптивными вариантами, что позволяет чрезвычайно легко создавать адаптивные интерфейсы, не прибегая к пользовательскому CSS.

- Tailwind использует интуитивно понятный *{screen}*: префикс, который позволяет легко заметить реагирующие классы в разметке, сохраняя при этом оригинальное имя класса узнаваемым и неповрежденным.

- Tailwind предоставляет инструменты для извлечения классов компонентов из повторяющихся шаблонов утилит, что позволяет легко обновлять несколько экземпляров компонента из одного места.

- Tailwind написан на PostCSS и настроен на JavaScript, что означает, что у вас есть все возможности настоящего языка программирования под рукой.

- Tailwind — это не просто CSS-фреймворк, это движок для создания систем проектирования.

Также в данной работе будут использоваться CSS и HTML.

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTML – язык разметки документов.

1.2.3. Выбор языков программирования

Наиболее популярными языками программирования для реализации веб-приложения считаются: Java, PHP, C# и JavaScript.

Для выполнения данной работы были использованы такие языки, как PHP и JavaScript [22].

JavaScript – самый известный язык программирования в разработке веб-приложений. JavaScript является объектно-ориентированным и дает возможность реализовать сложную механику веб-страниц.

Достоинства JavaScript:

- JavaScript может использоваться как на серверной стороне (backend), так и на клиентской (frontend);
- по умолчанию JavaScript поддерживается всеми распространенными браузерами и включен в них;
- полная интеграция с HTML/CSS.

PHP – скриптовый язык программирования, разработанный для написания веб-приложений. PHP является одним из первых языков для разработки веб-приложений и остается одним из самых популярных на рынке [10].

Достоинства PHP:

- ориентация на веб-разработку – PHP создавался и развивался как язык для создания Web-сайтов. Многие конструкции и решения в нем созданы для удобства работы в веб-среде;
- кроссплатформенность – PHP совместим со всеми основными платформами (Windows, Unix, Linux и MacOS), поддерживает большинство серверов (Apache, Microsoft IIS, Tornado, iPlanet, Xitami и другие) и более 20 БД;
- открытый доступ. Не нужно платить за получение лицензии;
- высокая гибкость и функциональность.

1.3.ВЫВОД

Федерация современного пятиборья Челябинской области и МБУ «Спортивная школа Конноспортивный комплекс Рифей им. П.М.Латышева» г. Челябинска нуждается в разработке единого программно-аппаратного комплекса для организации соревнований и тренировочного процесса, что позволит спортсменам и тренерам достигать максимальных результатов, ускорить процесс проведения соревнований и исключить нежелательные ошибки в процессе обработки информации.

Посредством веб-интерфейса система должна предоставлять следующие функции:

1. Ведение базы данных спортсменов, тренеров, организаторов соревнований.
2. Создание и редактирование соревнований.
3. Ведение личного кабинета спортсменов, тренеров, организаторов соревнований (судей).
4. Отслеживание динамики изменений личных параметров спортсмена и результатов по всех соревнованиях, в которых он участвовал.
5. Хранение результатов соревнований.
6. Получение промежуточных и итоговых протоколов соревнований.

Для разработки проекта были выбраны следующие программные компоненты:

- СУБД MySQL;
- язык программирования: PHP;
- для верстки: HTML, CSS, JavaScript;
- фреймворк TailwindCSS;
- текстовый редактор: PhpStorm с использованием локального сервера OpenServer со встроенным веб-приложением PhpMyAdmin.

2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации данного комплекса необходим следующий набор подсистем.

- Веб-приложение. Приложение обеспечивает доступ пользователей к комплексу.
- База данных MySQL.
- Аппаратная реализация комплекса.

2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Веб-приложение должно поддерживать следующую ролевую модель, а также возможность управления ролями и их правами.

Роли зарегистрированных пользователей: тренер, спортсмен, организатор.

Доступ к различным функциям должен разграничиваться в зависимости от того, какая роль присвоена пользователю.

Тренер должен иметь следующие возможности:

- осуществлять регистрацию/авторизацию в веб-приложении;
- редактировать профиль в личном кабинете;
- создавать календарный план тренировок и соревнований;
- просматривать новости и документы;
- создавать и редактировать тренировочные соревнования;
- просматривать текущие и итоговые результаты соревнований.

Спортсмен должен иметь следующие возможности:

- осуществлять регистрацию/авторизацию в веб-приложении;
- редактировать профиль в личном кабинете;

- отслеживать индивидуальные параметры (вес, рост, обхват груди/талии/бедер, индекс массы тела (ИМТ)) в личном кабинете;
- создавать и редактировать календарь тренировок и соревнований;
- просматривать все соревнования, в которых он участвовал ранее через личный кабинет;
- просматривать новости и документы;
- просматривать текущие и итоговые результаты соревнований.

Организатор должен иметь следующие возможности:

- создавать/редактировать/удалять официальные и тренировочные соревнования;
- добавлять спортсменов/судей в текущие соревнования;
- просмотр итоговых протоколов всех соревнований и промежуточных результатов текущих соревнований.

Администратор должен иметь доступ ко всем данным.

Регистрация пользователя включает в себя следующие поля: ФИО, пол, дата рождения, номер телефона, город, роль, email, пароль, повторение пароля.

В зависимости от выбора роли могут появляться дополнительные поля для заполнения:

- для спортсмена: выбор тренера, спортивный разряд;
- для тренера: тренерская категория, стаж работы и вид спорта;
- для организатора: судейская категория.

Подчеркнутые поля должны быть оформлены в виде выпадающего списка:

- пол: женский/мужской;
- роль: спортсмен, тренер, организатор;
- город: Челябинск, Москва, Нижний Новгород, Самара, Екатеринбург, Санкт-Петербург, Казань, Уфа, Ростов-на-Дону, Краснодар,

Калининград, Нальчик, Ставрополь, Киров, Новосибирск, Воронеж, Белгород, Владивосток;

- спортивный разряд: ЗМС, МСМК, МС, КМС, 1-3 взрослые разряды, 1-3 юношеские разряды;
- судейская категория: судья 1-3 категории, всероссийская категория;
- виды спорта: плавание, комбайн, конкур, фехтование;
- тренерская категория: первая, вторая, высшая.

Также у спортсменов есть возможность вносить в личном кабинете индивидуальные параметры, такие как: вес, рост, обхват груди/талии/бедер.

По двум из параметров (рост и вес) выводится текущее значение индекса массы тела (ИМТ), которое позволяет оценить, является ли масса спортсмена недостаточной, нормальной или избыточной. В зависимости от этого можно изменять рацион питания, режим тренировок и витаминизацию атлета.

2.2.НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.2.1. Общие требования

Веб-приложение должно быть частью программно-аппаратного комплекса, но при этом не зависеть от аппаратной части комплекса.

Невозможность прохождения процедуры регистрации без заполнения всех полей.

2.2.2. Требования к аппаратной части комплекса

Стандартную стрелковую мишень необходимо дооснастить средством беспроводной передачи данных (далее модуль сопряжения).

Модуль сопряжения должен:

- быть выполнен в виде отдельного конструктива с автономным питанием;
- быть способным принимать данные от мишени по UIPM Open target protocol и передавать их в Облако посредством Wi-Fi.

2.2.3. Требования к надежности

Для восстановления информации необходимо обеспечить автоматизированное резервное копирование по расписанию, установленному администратором системы. Копии должны храниться на сервере. Процедура регистрации невозможна без заполнения всех полей, также вся информации должна соответствовать требованиям поля.

2.2.4. Требования к документации

В документации на аппаратную часть комплекса должны содержаться технические характеристики, которые включают: вес, габариты устройства, тип источника питания, формат выходных данных, класс защиты корпуса, рабочая температура, тип лазерного сигнала.

Документация будет предоставляться заказчику в электронном виде. Документация должна быть на русском языке.

2.2.5. Требования к безопасности

Доступ ко всем данным осуществляется в соответствии с правами категорий пользователей.

Все пароли пользователей должны содержать буквы и цифры. Также пароли пользователей должны быть захешированы.

3. ПРОЕКТИРОВАНИЕ

3.1.АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

3.1.1. Архитектура аппаратной части комплекса

Стандартная стрелковая мишень регистрирует факт первого выстрела (вне зависимости от его качества), что является признаком начала стрелковой серии и контролирует время стрелковой серии (50 с). Качество выстрела отображается с помощью световой индикации, но время, потраченное на выполнение серии из пяти попаданий, нигде не регистрируется и не сохраняется.

Задача модуля сопряжения заключается в принятии данных о выстрелах, времени, затраченного на их выполнение и передачи этой информации в «Облачное хранилище» для дальнейшего использования.

На рисунке 3.1 показана структура программно-аппаратного комплекса для фиксации результатов со стрелковой мишени.

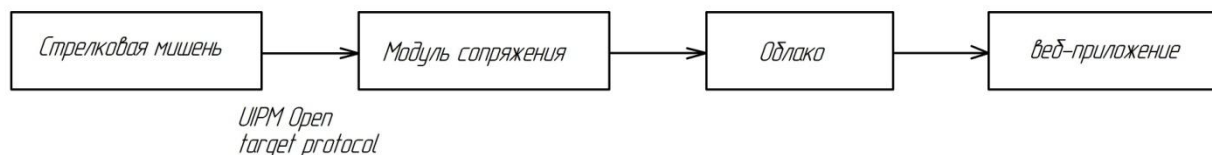


Рисунок 3.1 – Схема комплекса для фиксации результата со стрелковой мишени

Для реализации модуля сопряжения понадобится: микроконтроллер, модуль Wi-Fi, кнопки ВКЛ/ВЫКЛ и RESET, светодиодная индикация заряда и «POWER», источник питания, блок подзарядки аккумуляторной батареи.

Для возможности прошивки необходим разъем программатора. Структурная схема модуля сопряжения изображена на рисунке 3.2.

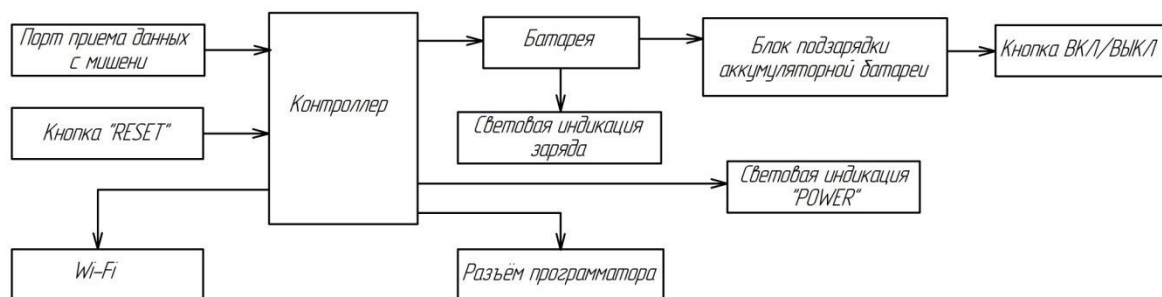


Рисунок 3.2 – Структурная схема модуля сопряжения

При проектировании модуля сопряжения необходимо выполнить подбор всех элементов, представленных на структурной схеме.

Микроконтроллер – микросхема, предназначенная для управления электронными устройствами. Типовой микроконтроллер сочетает в себе функции процессора и периферийных устройств, может содержать ОЗУ и ПЗУ. В качестве микроконтроллера выбран ATmega 8L (рисунок 3.3). Более подробно характеристики рассмотрены в [16].

ATmega 8 – это высокопроизводительный 8-битный микроконтроллер фирмы AVR. Характеристики микроконтроллера ATmega 8L:

- 32 8-битных рабочих регистра;
- внешняя тактовая частота до 16 МГц;
- ОЗУ на 8 кбайт;
- 512 байт EEPROM;
- 2 8-битных таймера/счетчика;
- 1 16-битный таймер/счетчик;
- программируемый сторожевой таймер;
- возможность обнаружения подачи и сбоя питания;
- 23 порта ввода/вывода;
- 6 аналоговых портов ввода/вывода;

- напряжение питания +3,3 В;
- рабочая температура: -40°C ... +85°C.

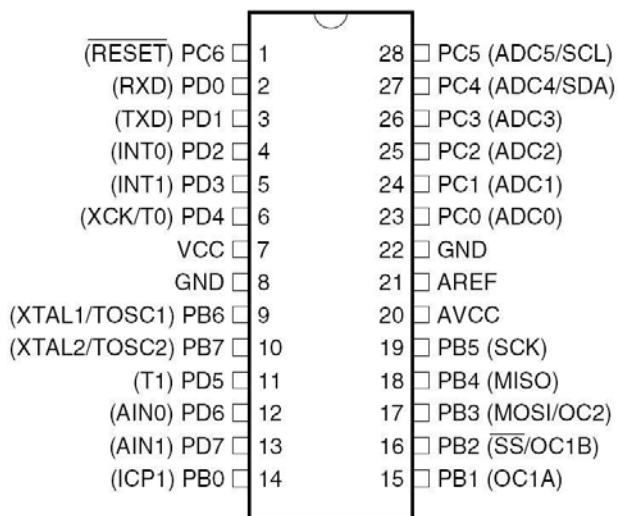


Рисунок 3.3 – Распиновка микроконтроллера АТмега8L

Схема подключения микроконтроллера АТмега 8L изображена на рисунке 3.4.

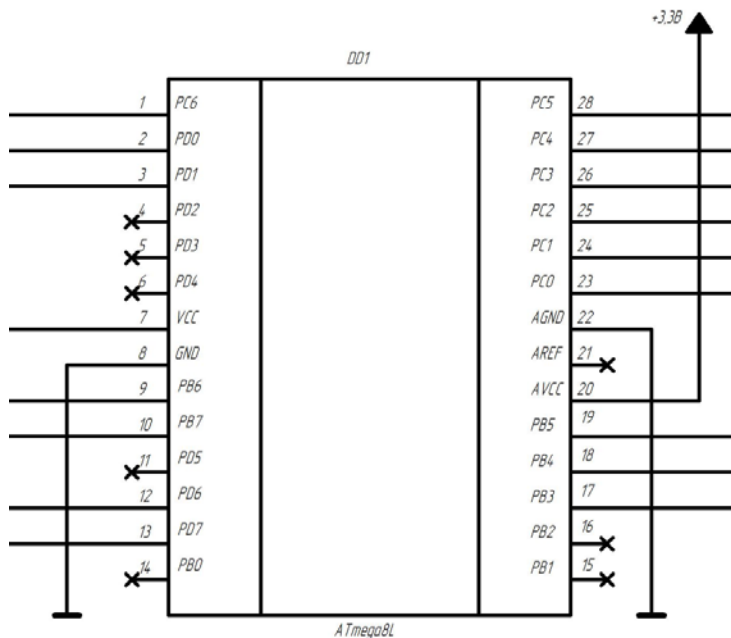


Рисунок 3.4 – Микроконтроллер АТмега 8L

Портом приема данных со стрелковой мишени выступает разъем RJ-45 (Ethernet) (рисунок 3.5).

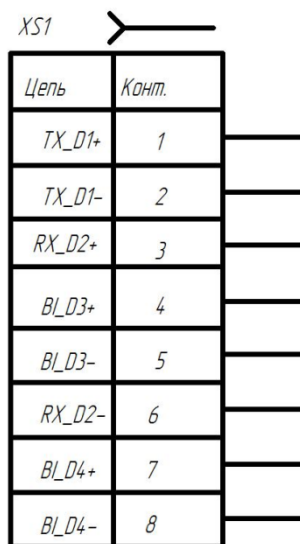


Рисунок 3.5 – Разъем подключения микроконтроллера к мишени (RJ-45)

ESP8266 представляет собой микроконтроллер с интерфейсом Wi-Fi, который имеет возможность исполнять программы из Flash-памяти. Устройство было выпущено в 2014 году фирмой Espressif (КНР) и сразу стало популярным [15]. Модуль ESP8266 изображен на рисунке 3.6.

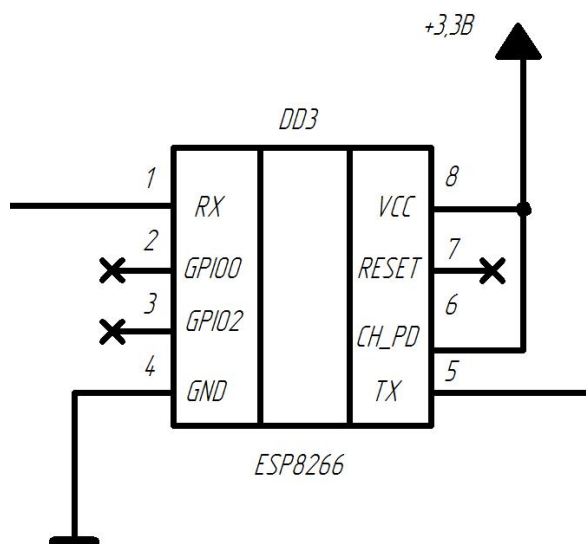


Рисунок 3.6 – Модуль Wi-Fi ESP8266

Технические характеристики модуля ESP8266:

- напряжение питания 3,3 В;
- в режиме передачи - ток до 215 мА;
- во время приема - ток до 62 мА;
- поддерживает Wi-Fi протоколы: 802.11 b/g/n с WEP, WPA, WPA2;
- SDIO (два вывода присутствуют на плате модуля);
- режимы сохранения энергии и сна для экономии энергии;
- встроенный микроконтроллер;
- управление посредством AT-команд;
- температура функционирования от -40 до +125°C;
- максимальная дистанция связи 100 м;
- тактовая частота процессора - 80 МГц.

Программа выполняется от внешней SPI ПЗУ при помощи динамической загрузки необходимых элементов программы. Доступ к внутренней периферии можно получить не из документации, а из API набора библиотек.

Описание контактов модуля Wi-Fi ESP8266:

- GND - «земля»;
- VCC – питание;
- RESET – необходим для перезагрузки микроконтроллера при подаче на него низкого логического уровня;
- CP_PD – используется для перевода устройства в энергосберегающий режим;
- RXD0 и TXD0 – аппаратный UART, необходимый для прошивки модуля;

- GPIO0 – порт ввода и вывода, позволяющий перевести устройство в режим программирования (при подключении порта к низкому логическому уровню и подачи напряжения);
- GPIO2 – порт ввода и вывода.

Кнопка «RESET» используется в процессе разработки устройства для сброса микроконтроллера в начальное состояние.

Чтобы обеспечить повторный запуск микроконтроллера при отладке устройства необходимо сделать возможность выполнения сброса, а также его надежный сброс при включении питания (цепочка R1-C3).

Схема подключения кнопки Reset к микроконтроллеру изображена на рисунке 3.7.

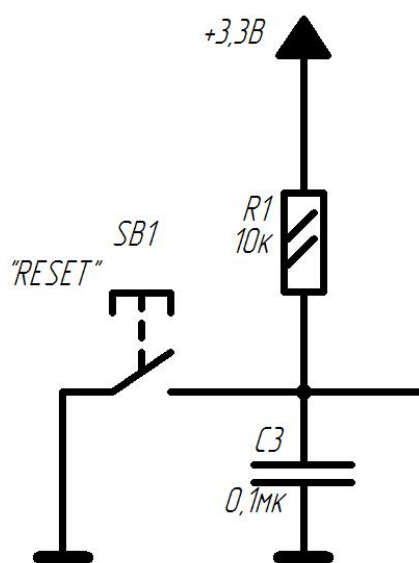


Рисунок 3.7 – Подключение кнопки «RESET» к микроконтроллеру

Тактовая частота, на которой работает микроконтроллер, определяет скорость его работы.

Каноническим способом тактирования микроконтроллера является подключение кварцевого резонатора. Емкость конденсаторов С1 и С2 в типовом случае должна составлять 22-36 пФ [14].

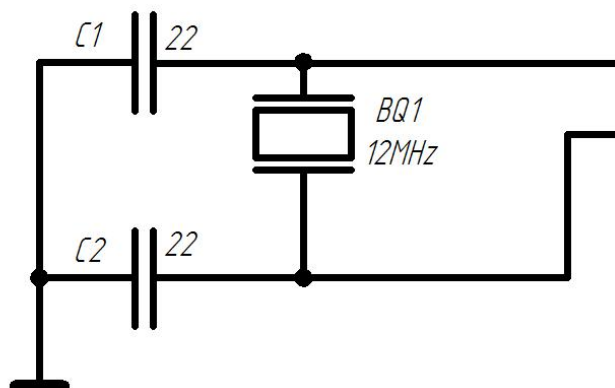


Рисунок 3.8 – Кварцевый резонатор

Для тактирования нужно подключить кварцевый резонатор по схеме из технической документации на микроконтроллер (RB6 и RB7). Схема подключения показана на рисунке 3.8.

Программатор представляет собой программно-аппаратное устройство, которое используют для занесения необходимой информации в постоянное запоминающее устройство (ПЗУ). Также программатор обеспечивает возможность считывания информации из постоянного запоминающего устройства микроконтроллера.

Существует два варианта подключения программатора к микроконтроллеру:

- программирование происходит непосредственно в колодке программатора;
- внутрисхемно.

Первый вариант подключения возможен только при использовании DIP-корпусов.

Для внутрисхемного программирования используются специальные разъемы для подключения программатора [13].

Главное преимущество внутрисхемного программирования: исключение отдельной фазы программирования микросхем перед окончательной сборкой устройства с помощью объединения процесса программирования и тестирования.

Как правило, подобные микросхемы имеют встроенные схемы, которые:

- генерируют из внешнего напряжения питания напряжения, необходимые для программирования;
- обеспечивают коммуникацию с внутрисхемным программатором по последовательному интерфейсу (вариации протокола JTAG, SPI, UART).

Схема подключения разъема программатора к микроконтроллеру изображена на рисунке 3.9.

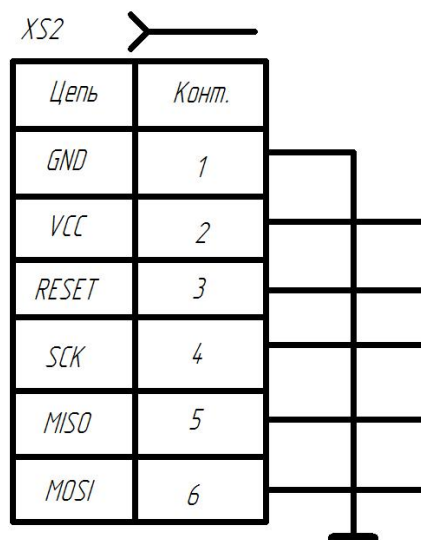


Рисунок 3.9 – Разъем программатора

Подключение программатора к устройству: для того, чтобы подключить программатор к микроконтроллеру используется три линии интерфейса, линия

сброса (Reset), плюс и минус питания (VCC и GND). К линиям интерфейса относятся тактовый сигнал (SCK), вход данных (MOSI) и выход данных (MISO).

Посылая 4-байтные команды на вывод MOSI микроконтроллера, осуществляется программирование, а снятие результатов выполнения команд чтения снимается с MISO.

На рисунке 3.10 изображен источник питания модуля сопряжения.

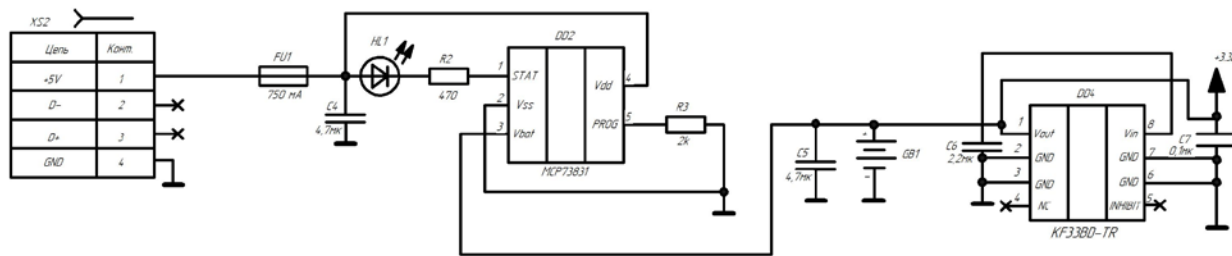


Рисунок 3.10 – Источник питания

Для возможности подзарядки устройства от USB компьютера или другого универсального блока питания с микро USB используем разъем микро USB. Схема подключения разъема изображена на рисунке 3.11.

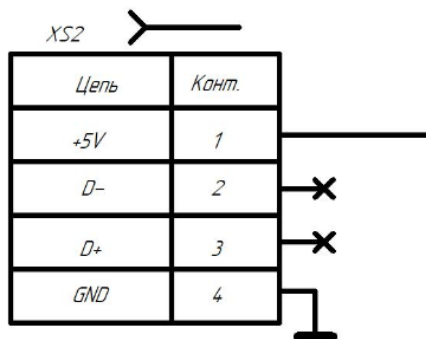


Рисунок 3.11 – Разъем микро USB

Рассмотрим подключение контроллера заряда, светодиода и предохранителя к микроконтроллеру (рисунок 3.12).

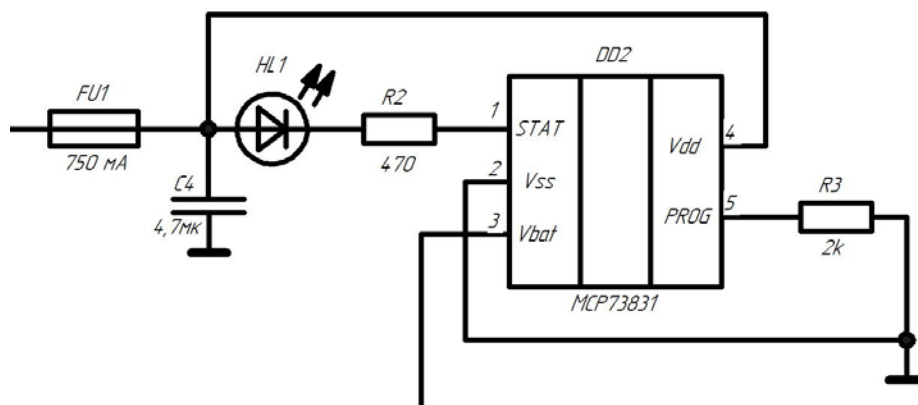


Рисунок 3.12 – Подключение контроллера заряда, светодиода и предохранителя

Контроллер заряда – это устройство, которое автоматически регулирует уровень тока и напряжения от источника для обеспечения заряда аккумуляторных батарей (АКБ), таким образом, предохраняя аккумуляторы от повреждений. Для реализации комплекса выберем контроллер заряда MCP73831.

Характеристики контроллер заряда MCP73831:

- входное напряжение: 6 В;
- напряжение заряда батареи 4,2 В;
- минимальный ток зарядки: 500 мА;
- тип батареи: литий-ионная, литий-полимерная (Li-Ion, Li-Pol);
- количество выводов: 8 выводов;
- размеры: 3·2·0,95мм;
- рабочая температура: -40°С ... +85°С.

Более подробно технические характеристики рассмотрены в [19].

Для индикации заряда предусмотрена световая индикация. Выберем светодиод КР-1608MGC. На рисунке 3.12 светодиод обозначен буквами HL1. В таблице 3.1 приведены характеристики светодиода КР-1608MGC. Более подробно характеристики рассмотрены в [17].

Таблица 3.1 – Характеристики светодиода KP-1608MGC

Ток через диод, мА	30
Падение напряжения, В	2,2
Максимальная сила света, мКд	70
Цвет свечения	Зеленый
Типоразмер	0603

Предохранитель – электрический аппарат, предназначенный для защиты электрических цепей и ее элементов от перегрева и возгорания при протекании тока высокой силы [18].

Предохранитель является эклектическим аппаратом, главной функцией которого выступает защита цепей и ее элементов от возгораний/перегрева при протекании тока высокой силы.

На рисунке 3.12 предохранитель обозначен буквами FU1 (от английского «Fuse») в виде прямоугольника со сплошной линией в центре. Нам понадобится предохранитель с номинальным рабочим током равным 750 мА.

Стабилизатор напряжения – электрическое устройство со входом и выходом по напряжению, которое позволяет при существенном изменении входного напряжения и выходного тока нагрузки поддерживать выходное напряжение в узких пределах.

Главным назначением стабилизатора является защита электрических устройств от возможного повреждения в результате скачка напряжения или колебаний, повышенного или пониженного напряжения [20].

В качестве стабилизатора напряжения выберем KF33BD-TR, выходное напряжение которого 3,3 В. Схема подключения стабилизатора напряжения изображена на рисунке 3.14.

Характеристики стабилизатора напряжения KF33BD-TR:

- полярность: положительная;
- тип входа: фиксированный;
- выходное напряжение: 3,3 В;
- максимальный ток нагрузки: 0,5 А;
- максимальное входное напряжение: 20 В;
- рабочая температура: -40...+125°C.

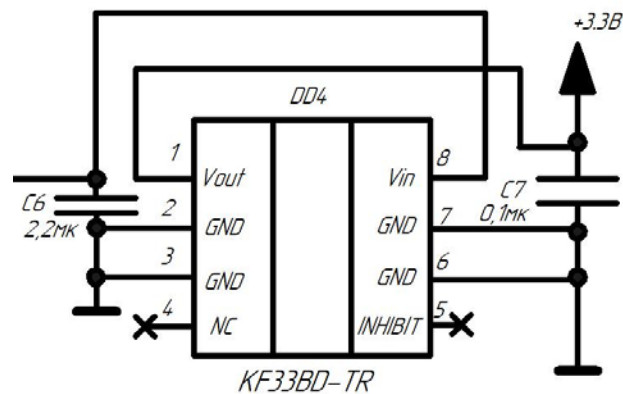


Рисунок 3.14 – Стабилизатор напряжения KF33BD-TR

Общая принципиальная схема модуля сопряжения изображена на рисунке 3.15.

3.1.2. Архитектура программной части комплекса

Веб-приложение является клиент-серверным: «клиентом» выступает браузер, а «сервером» - веб-сервис.

Архитектуру «клиент-сервер» можно поделить на двухзвенную, трехзвенную и многозвенную.

Двухзвенная архитектура используется в системах, где сервер отвечает на клиентские запросы напрямую (рисунок 3.16).

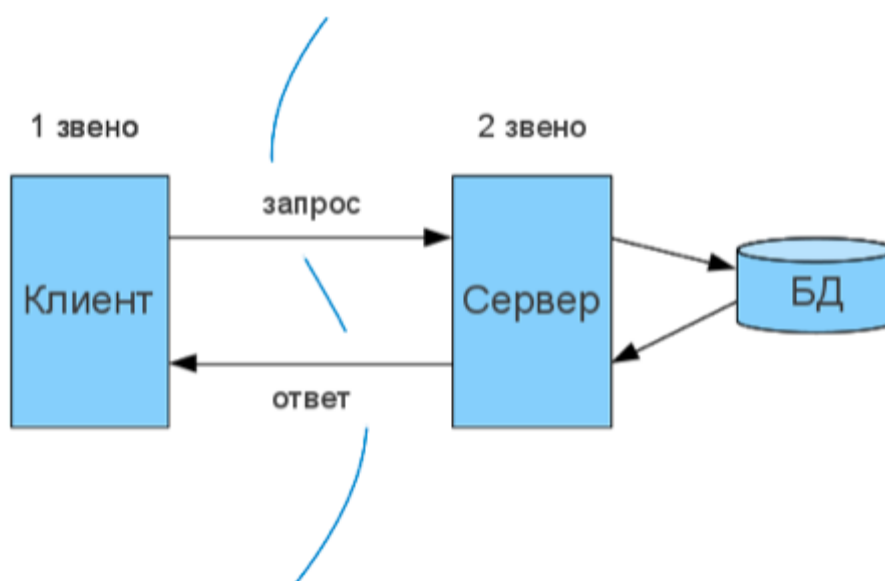


Рисунок 3.16 – Двухзвенная клиент-серверная архитектура

Основные модели взаимодействия компонентов на стороне сервера и клиента в данной архитектуре:

- сервер терминалов – распределенное представление данных;
- файл-сервер – доступ к удаленной базе данных и файловым ресурсам;
- сервер БД – удаленное представление данных;
- сервер приложений – удаленное приложение.

В двухзвенной архитектуре в качестве клиента выступает браузер. Для того чтобы пользователь увидел графический интерфейс приложения в окне браузера,

ему необходимо обработать полученный ответ веб-сервера, в котором храниться вся информация, реализованная с применением HTML и CSS. Данные технологии позволяют браузеру определить, как конкретно отобразить все, что он получил в ответе.

Веб-сервер – это сервер, который принимает HTTP-запросы от клиентов и выдает им HTTP-ответы вместе с различными изображениями и другими файлами.

База данных позволяет упорядочить хранимые данные об объектах, обладающих различными свойствами.

Добавив к двухзвенной архитектуре сервер приложений можно получить трехзвенную архитектуру (рисунок 3.17). Распределение компонентов в трехзвенной модели:

1. Представление данных – сторона клиента (браузера).
2. Прикладной компонент – сервер приложений.
3. Управление ресурсами – сервер БД, представляющий запрашиваемые данные.

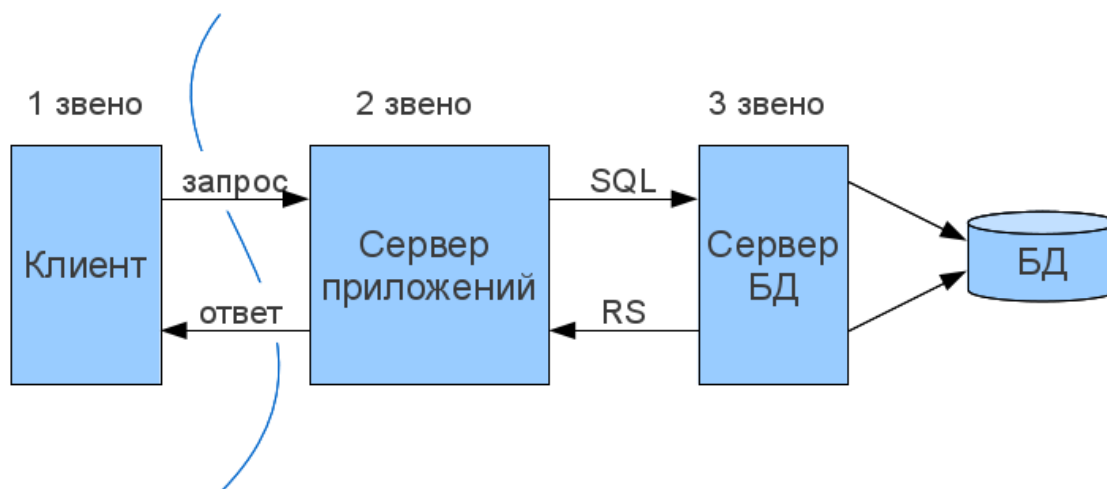


Рисунок 3.17 – Трехзвенная клиент-серверная архитектура

Трехзвенная архитектура может быть расширена многозвенной путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня (рисунок 3.18).

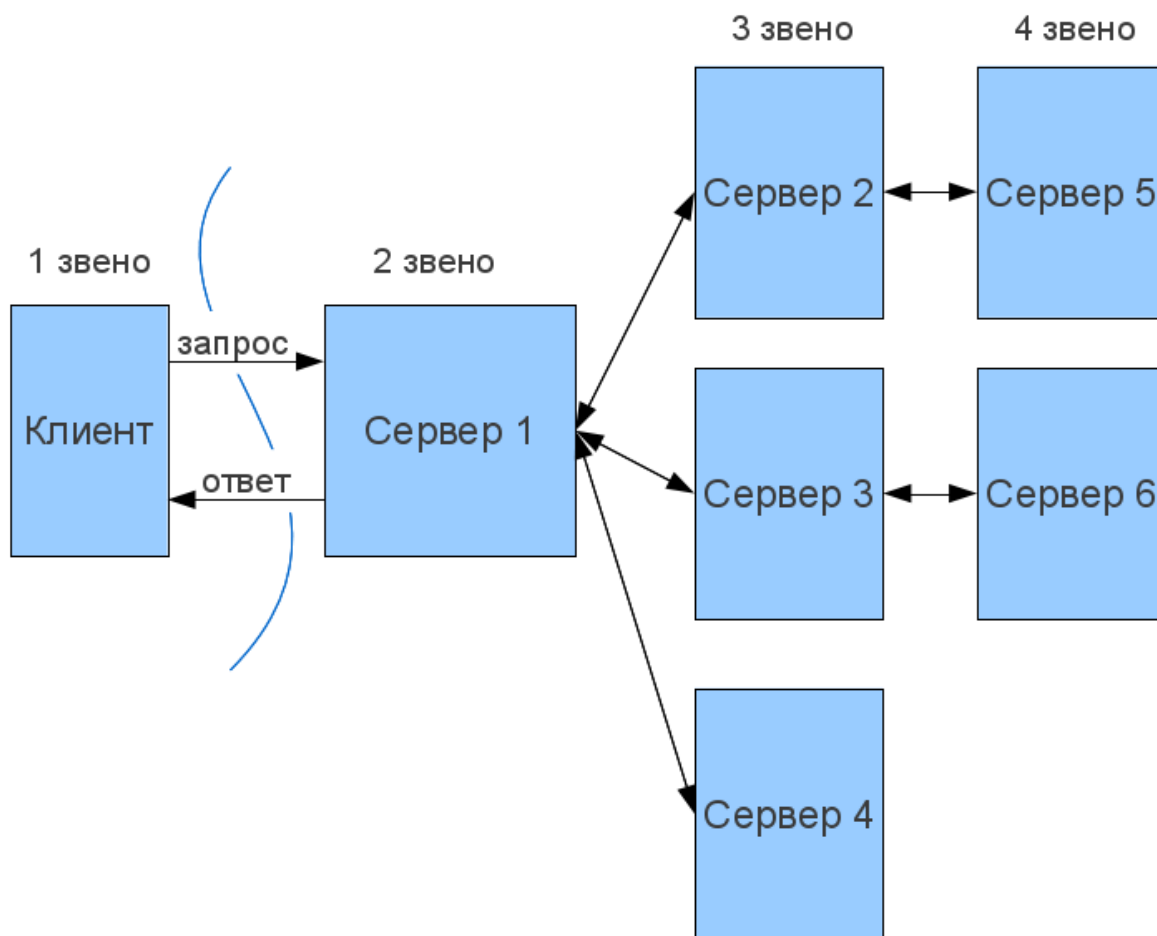


Рисунок 3.18 – Многозвенная клиент-серверная архитектура

Главными преимуществами данной архитектуры являются высокая безопасность и производительность.

Для доступа к сетевым сервисам используются клиенты, возможности которых характеризуются понятием «толщины». Рассмотрим различные виды клиентов.

«Тонкий» клиент

Этот термин определяет клиента, вычислительных ресурсов которого достаточно лишь для запуска необходимого сетевого приложения через web-интерфейс. Пользовательский интерфейс такого приложения формируется средствами статического HTML, вся прикладная логика выполняется на сервере. Для работы тонкого клиента достаточно лишь обеспечить возможность запуска web-браузера, в окне которого и осуществляются все действия. По этой причине web-браузер часто называют «универсальным клиентом».

«Толстый» клиент

Представляет собой рабочую станцию или персональный компьютер, работающий под управлением собственной дисковой операционной системы и имеющий необходимый набор программного обеспечения. К сетевым серверам «толстые» клиенты обращаются за дополнительными услугами (например, доступ к web-серверу или корпоративной базе данных).

Так же под «толстым» клиентом подразумевается и клиентское сетевое приложение, запущенное под управлением локальной ОС. Такое приложение совмещает компонент представления данных (графический пользовательский интерфейс ОС) и прикладной компонент (вычислительные мощности клиентского компьютера).

«Rich» клиент

«Rich» клиент является компромиссом между «толстым» и «тонким» клиентом. Как и «тонкий» клиент, «rich»-клиент также представляет графический интерфейс, описываемый уже средствами XML и включающий некоторую функциональность толстых клиентов (например, интерфейс drag-and-drop, вкладки, множественные окна, выпадающие меню и т.п.)

Прикладная логика «rich»-клиента также реализована на сервере. Данные отправляются в стандартном формате обмена.

Вывод: для реализации программной части комплекса будет использоваться трехфазная архитектура «клиент-сервер». Для проектирования архитектуры будет использован «тонкий» клиент, так как большая часть задач по обработке информации перенесена на сервер приложения и права доступа клиентов строго ограничены.

3.2. ОПИСАНИЕ ДАННЫХ

Схема базы данных веб-приложения, реализованной в СУБД MySQL, изображена на рисунке 3.19.

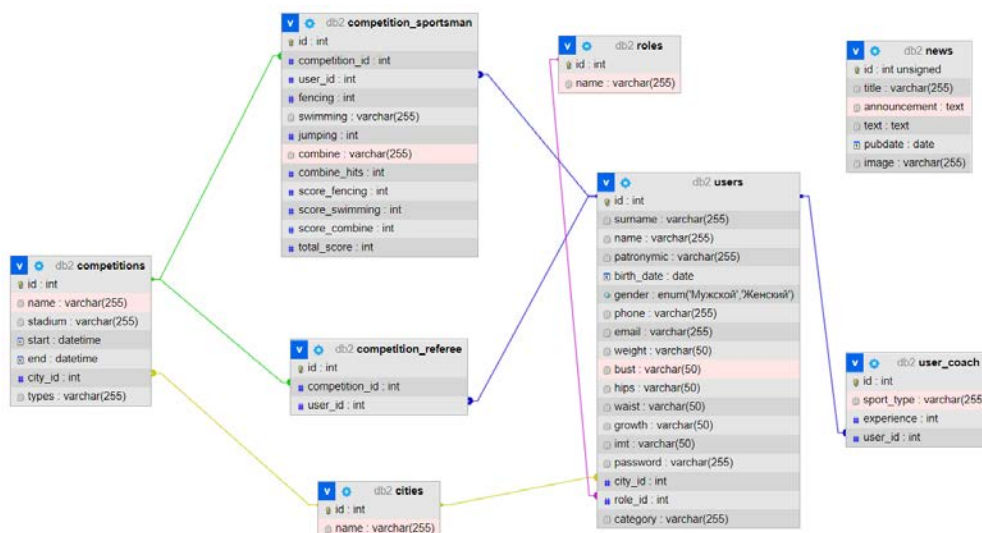


Рисунок 3.19 – Схема базы данных

В таблице 3.2 представлен список таблиц базы данных.

Таблица 3.2 – Список таблиц базы данных

Название таблицы	Назначение
Users	Данные о пользователях
Roles	Список ролей пользователей

Продолжение таблицы 3.2

Cities	Список городов
Competitions	Данные о соревнованиях
Competition_referee	Связь между организаторами и соревнованиями
Competition_sportsman	Связь между спортсменами и соревнованиями
User_coach	Данные о тренерах
News	Новости

В таблице 3.3 описаны поля таблицы «Users».

Таблица 3.3 – Поля таблицы «Users»

Наименование	Тип данных	Назначение
id	int(11)	Уникальный идентификатор
surname	varchar(255)	Фамилия
name	varchar(255)	Имя
patronymic	varchar(255)	Отчество
bith_date	date	Дата рождения
gender	enum('Мужской', 'Женский')	Пол
phone	varchar(255)	Телефон
email	varchar(255)	Электронная почта
weight	varchar(50)	Масса тела (Вес)
bust	varchar(50)	Обхват груди (ОГ)
hips	varchar(50)	Обхват бедер (ОБ)
waist	varchar(50)	Обхват талии (ОТ)
growth	varchar(50)	Рост
imt	varchar(50)	Индекс массы тела (ИМТ)

Продолжение таблицы 3.3

Наименование	Тип данных	Назначение
password	varchar(255)	Пароль
city_id	int(11)	Идентификатор города
role_id	int(11)	Идентификатор роли
category	varchar(255)	Судейская категория

В таблице 3.4 описаны поля таблицы «Roles».

Таблица 3.4 – Поля таблицы «Roles»

Наименование	Тип данных	Назначение
id	int(11)	Уникальный идентификатор
name	varchar(255)	Название роли

В таблице 3.5 описаны поля таблицы «Cities».

Таблица 3.5 – Поля таблицы «Cities»

Наименование	Тип данных	Назначение
id	int(11)	Уникальный идентификатор
name	varchar(255)	Название города

В таблице 3.6 описаны поля таблицы «Competitions».

Таблица 3.6 – Поля таблицы «Competitions»

Наименование	Тип данных	Назначение
id	int(11)	Уникальный идентификатор
name	varchar(255)	Название
stadium	varchar(255)	Стадион
start	datetime	Начало
end	datetime	Конец
city_id	int(11)	Идентификатор города
types	varchar(255)	Тип соревнования

В таблице 3.7 описаны поля таблицы «Competition_referee».

Таблица 3.7 – Поля таблицы «Competition_referee»

Наименование	Тип данных	Назначение
id	int(11)	Уникальный идентификатор
competition_id	int(11)	Идентификатор соревнования
user_id	int(11)	Идентификатор пользователя

Таблица 3.8 – Поля таблицы «Competition_sportsman»

Наименование	Наименование	Назначение
id	int(11)	Уникальный идентификатор
competition_id	int(11)	Идентификатор соревнования
user_id	int(11)	Идентификатор пользователя
fencing	varchar(255)	Результат фехтования
swimming	varchar(255)	Результат плавания
jumping	varchar(255)	Результат конкура
combine	varchar(255)	Результат комбайна
combine_hits	int(11)	Стрельба
score_fencing	int(11)	Баллы за фехтование
score_swimming	int(11)	Баллы за плавание
score_combine	int(11)	Баллы за комбайн
total_score	int(11)	Общие баллы

В таблице 3.9 описаны поля таблицы «User_coach».

Таблица 3.9 – Поля таблицы «User_coach»

Наименование	Наименование	Наименование
id	int(11)	Уникальный идентификатор
sport_type	varchar(255)	Вид спорта
experience	int(11)	Стаж работы
user_id	int(11)	Идентификатор пользователя

В таблице 3.10 описаны поля таблицы «News».

Таблица 3.10 – Поля таблицы «News»

Наименование	Наименование	Наименование
id	int(11)	Уникальный идентификатор
title	varchar(255)	Заглавие новости
announcement	text	Анонс новости
text	text	Текст новости
pubdate	date	Дата публикации
image	varchar(255)	Изображение

4. РЕАЛИЗАЦИЯ

Балльно-рейтинговая система

Рассмотрим начисление баллов в современном пятиборье. Общее количество баллов по многоборью вычисляется по сумме баллов, набранных за выполнение каждого вида спорта (плавание, фехтование, конкур и комбинированный вид).

Первоначальным в плавании считается время 2:30 с, которое приравнивается к 250 баллам. Каждые 0,5 с выше (ниже) данного результата улучшают (ухудшают) его показатель на 1 балл.

Пример: результат спортсмена 2: 35 с = $250 - 10 = 240$ баллов.

В фехтовании 70% выигранных боев расцениваются как 250 баллов. За каждые уколы выше (ниже) данного результата улучшают (ухудшают) данный результат на 8 баллов.

Пример: если в соревновании принимают участие 24 спортсмена, то 70% составляет 16 уколов. Участник нанес 18 уколов (побед), значит, его результат составляет $250 + 2 * 8 = 266$ баллов.

В конкуре у каждого участника соревнований первоначально имеется 300 баллов. Они начисляются, если дистанция пройдена без штрафов и за время 1:10 с и менее. За нарушения различного рода снимается от 7 до 10 баллов штрафа. Также по конкуру можно получить 0 баллов за совершение серьезных нарушений, таких как: жестокое обращение с животными, 3 неудачных попытки захода на барьер, неуважительное отношение к организаторам соревнований, нарушение регламента, нарушение маршрута дистанции и т.д.

В комбайне первоначальным считается время 11:40 с, которое приравнивается в 600 баллам. Каждая 1 с выше (ниже) этого результата улучшает (ухудшает) его показатель на 1 балл.

Например: результат спортсмена по комбайн $11:50 \text{ с} = 600 - 10 = 590$ баллов.

В листинге А.5 приложения А содержится исходный код функции вычисления баллов по каждому виду современного пятиборья.

Форма регистрации пользователя

Регистрация пользователя включает в себя следующие поля: ФИО, пол, дата рождения, номер телефона, город, роль, email, пароль.

В зависимости от выбранной роли появляются дополнительные поля для заполнения:

- для спортсмена: спортивный разряд;
- для тренера: тренерская категория, стаж работы и вид спорта;
- для организатора: судейская категория.

Поля Пол, Город, Моя роль, Спортивный разряд, Тренерская категория, Вид спорта и Судейская категория имеют выпадающий список для выбора одного из возможных вариантов заполнения. Возможные варианты:

- пол: женский/мужской;
- моя роль: тренер, организатор, спортсмен;
- спортивный разряд: ЗМС, МСМК, МС, КМС, 1-3 взрослые разряды, 1-3 юношеские разряды;
- судейская категория: судья 1-3 категории, всероссийская категория;
- виды спорта: плавание, комбайн, конкур, фехтование;
- тренерская категория: первая, вторая, высшая.

Поля, которые являются обязательные для заполнения, отмечены в форме символом *. Форма регистрации представлена на рисунке 4.1.

В листинге А.1 приложения А содержится исходный код регистрации пользователей.

Регистрация

Фамилия*

Имя*

Отчество

Дата рождения*

Город*

Пол*

Электронная почта*

Телефон*

Пароль*

Повторите пароль еще раз*

Моя роль*

Разряд*

[Уже зарегистрированы? Войти](#)

Рисунок 4.1 – Экранная форма регистрации пользователя

После успешной регистрации пользователь перенаправляется на страницу своего личного кабинета (рисунок 4.2).

Если пользователь имеет роль Спортсмен, то в его личном кабинете появляются дополнительные поля для заполнения, которые представляют из себя индивидуальные параметры, такие как: вес, рост, обхват груди/талии/бедер.

По двум из параметров (рост и вес) вычисляется текущее значение индекса массы тела (ИМТ), которое позволяет оценить, является ли масса спортсмена недостаточной, нормальной или избыточной. Значения ИМТ представлены на рисунке 4.3.

Также в личном кабинете спортсмена можно увидеть список 10 самых актуальных тренировочных и соревновательных мероприятий, которые были или будут проведены по современному пятиборью.

Ниже под персональными данными и календарем соревнований можно ознакомиться с результатами за те соревнования, в которых спортсмен участвовал ранее.

Личный кабинет

[Изменить профиль](#)

Фамилия: Баташова
 Имя: Ульяна
 Отчество: Евгеньевна
 Дата рождения: 11.11.1994
 Пол: Женский
 Телефон: +7 988 555 55 54
 Почта: ba@sport.ru
 Город: Москва
 Роль: Спортсмен
 Рост: 177 см
 Вес: 60.2 кг
 Обхват груди: 67 см
 Обхват талии: см
 Обхват бедер: см
 ИМТ: 19
 Разряд: ЗМС

Название	Город	Начало	Тип
Тренировочное соревнование Самара	Самара	25.04.2022 08:00:00	Тренировочные
Чемпионат России у женщин	Москва	20.03.2022 08:15:00	Официальные

Соревнование	Фехтование	Баллы за фехтование	Плавание	Баллы за плавание	Конкур	Комбайн	Стрелковые попадания	Баллы за комбайн	к
Тренировочное соревнование Самара	11	274	02:09	292	293	10:12	19	688	
Чемпионат России у женщин	9	266	02:09	292	293	10:12	20	688	

Рисунок 4.2 – Личный кабинет пользователя (роль - Спортсмен)

Индекс массы тела	Соответствие между массой человека и его ростом
16 и менее	Выраженный дефицит массы тела
16—18,5	Недостаточная (дефицит) масса тела
18,5—25	Норма
25—30	Избыточная масса тела (предожирение)
30—35	Ожирение
35—40	Ожирение резкое
40 и более	Очень резкое ожирение

Рисунок 4.3 – Значения ИМТ

В листинге А.2 приложения А содержится исходный код личного кабинета пользователей.

Форма авторизации пользователя

На рисунке 4.4. представлена форма входа для ранее зарегистрированных пользователей. Для успешной авторизации необходимо заполнить все поля для проверки корректности вводимых данных.

СОВРЕМЕННОЕ ПЯТИБОРЬЕ СОРЕВНОВАНИЯ О СПОРТЕ ДОКУМЕНТЫ ВХОД

Войдите в аккаунт, чтобы продолжить

Электронная почта

Пароль

Войти

[Нет аккаунта? Зарегистрируйтесь](#)

Рисунок 4.4 – Экранная форма авторизации пользователя

Форма редактирования профиля пользователя

Форма изменения личного профиля пользователя представлена на рисунке 4.5.

Изменение профиля

Фамилия*

Имя*

Отчество

Дата рождения*

Город*

Пол*

Электронная почта*

Телефон*

Категория*

Пароль* (Вписывать значение только, если Вы решили сменить пароль)

Повторите пароль еще раз*

Рисунок 4.5 – Экранная форма изменения личного профиля пользователя
(Роль - Организатор)

Форма создания новых соревнований

Для того чтобы просмотреть все существующие соревнования пользователю необходимо перейти на страницу «Соревнования». Если у пользователя роль Организатор (судья), то у него есть возможность добавить новое официальное или тренировочное соревнование. Для этого надо нажать на кнопку «Добавить соревнование» (рисунок 4.6), после чего появится форма добавления соревнования (рисунок 4.7). В данной форме необходимо заполнить следующие поля: Название соревнования, Город, Стадион проведения, Начало, Конец и выбрать судей, которые будут проводить данный старт. Выбор города оформлен в виде выпадающего списка.

У пользователя с ролью Тренер есть возможность создавать только тренировочное соревнование.

В листинге А.3 приложения А содержится исходный код страницы «Соревнования».

СОВРЕМЕННОЕ ПЯТИБОРЬЕ	СОРЕВНОВАНИЯ	О СПОРТЕ	ДОКУМЕНТЫ	ПРОФИЛЬ	ВЫЙТИ		
<hr/>							
Соревнования							
Добавить соревнование							
№	Название	Город	Стадион	Начало	Конец	Тип	
1	Тренировочное соревнование Самара	Самара	Центральный	25.04.2022 08:00:00	26.04.2022 19:30:00	Тренировочные	✗
2	Чемпионат России у женщин	Москва	Водный	20.03.2022 08:15:00	24.03.2022 20:15:00	Официальные	✗
<hr/>							

Рисунок 4.6 – Страница «Соревнования» (роль - Организатор)

Добавление соревнования

Название*

Город*

Стадион*

Начало*

Конец*

Судьи*

- Иванов Петр Иванович
- Непалов Игорь Алексеевич
- Терехова Светлана Семеновна
- Тестовой Сергей Алексеевич

Рисунок 4.7 – Экранная форма создания нового соревнования

Если же на страницу «Соревнование» перейдут пользователи с ролью Спортсмен, то она будет без кнопки «Добавить соревнование» и возможности удалить выбранное соревнование (рисунок 4.8).

Если на эту же страницу перейдет пользователи с ролью Тренер, то она будет иметь кнопку «Добавить соревнование», но при нажатии на нее будет автоматически указываться тип соревнования - Тренировочные (рисунок 4.9).

Соревнования

№	Название	Город	Стадион	Начало	Конец
1	Кубок России (Женщины)	Москва	Лужники	24.06.2020 08:00:00	01.07.2020 17:00:00
2	test	Новосибирск	Москва	24.05.2020 19:23:00	24.05.2020 19:22:00
3	Чемпионат России	Санкт-Петербург	Водный	02.04.2020 08:00:00	03.04.2020 17:50:00
4	Кубок России_2020 (Женщины)	Екатеринбург	Водный	22.01.2020 07:50:00	06.01.2020 16:55:00
5	Соревнование Новогоднее	Нижний Новгород	Звезда	07.01.2020 08:00:00	15.01.2020 15:00:00

Рисунок 4.8 – Экранная форма списка соревнований (роль - Спортсмен)

Добавление соревнования

Название *

Тип соревнований

Тренировочные

Тренировочные

Город*

Екатеринбург

Стадион*

Начало*

ДД.ММ.ГГГГ --:--

Конец*

ДД.ММ.ГГГГ --:--

Судьи*

- Иванов Петр Иванович
- Непалов Игорь Алексеевич
- Терехова Светлана Семеновна
- Тестовой Сергей Алексеевич
- Судья Для Тренировок

Рисунок 4.9 – Экранная форма добавления нового соревнования (роль - Тренер)

Форма добавления нового спортсмена в соревнование

После создания формы соревнования можно выбрать спортсменов, которые будут участвовать в новом соревновании. Форма добавления спортсменов в соревнование представлена на рисунке 4.10. Спортсмены выбираются из списка зарегистрированных пользователей с данной ролью.



Рисунок 4.10 – Экранная форма добавления спортсменов в соревнование

Форма редактирования соревнований

Если пользователь с ролью Организатор (судья) выберет конкретное соревнование, то появится возможность его отредактировать. Редактирование выглядит как таблица со следующими столбцами: номер, ФИО, Город и Год рождения участника соревнований; результаты по четырем видам многоборья (фехтование, плавание, конкур и комбайн) с количеством баллов, заработанных

за каждый вид; Общее количество баллов. Также есть возможность удалить участника соревнований. Форма редактирования соревнований представлена на рисунке 4.11.

[СОВРЕМЕННОЕ ПЯТИБОРЬЕ](#)
 [СОРЕВНОВАНИЯ](#)
 [О СПОРТЕ](#)
 [ДОКУМЕНТЫ](#)
 [ПРОФИЛЬ](#)
 [ВЫЙТИ](#)

Соревнование Новогоднее

№	ФИО	Город	Год рождения	Фехтование	Плавание	Конкур	Комбайн	Общие баллы	
1	Федеров Сергей Семенович	Челябинск	2002	0 (3)	0	17	780 (08:40)	797	✗
2	Камылев Игорь Сергеевич	Нижний Новгород	2000	0 (3)	250 (02:30)	2	0	252	✗

Рисунок 4.11 – Экранная форма редактирования соревнований

В листинге А.4 приложения А содержится исходный код добавления результатов в новое соревнование.

Главная страница и меню веб-приложения

Главная страница веб-приложения включает в себя главное меню, слайдер из пяти изображений с видами современного пятиборья и новостной ленты (рисунок 4.12).



Кубок Леднёва взял Полозков, чемпионат России – Громадский.

Сразу два победителя оказалось 27 мая на соревнованиях мужчин, завершившихся в московском районе «Северный». Второй этап Кубка Павла Леднёва выиграл белорусский пятиборец Илья Полозков, а в чемпионате России, который проходил в рамках этого турнира, отличился Егор Громадский. Еще один комплект высших наград, за победу в командном первенстве, вручили атлетам из сборной Москвы.



ЧР-2022. Самойлова и Буряк стали лучшими в полуфиналах у женщин

На проходящем в Москве Открытом чемпионате России, который также является вторым этапом Кубка Павла Леднёва, прошли предварительные соревнования у женщин. Лучшие результаты в своих группах показали россиянки Мария Самойлова и Анна Буряк (на фото).

Рисунок 4.12 – Главная страница веб-приложения

Также можно перейти на конкретную новость для более подробного ее изучения (рисунок 4.13).

Кубок Леднёва взял Полозков, чемпионат России – Громадский.

27.05.2022



В полуфинале белорус Илья Полозков уже после двух видов, плавания и фехтования, практически обеспечил себе первое место в группе, добившись весомого преимущества над соперниками. Похоже, что такой вариант развития событий пришелся ему по вкусу, и в финале он начал выступления по такой же схеме. В плавании стал четвертым, а фехтование выиграл, опередив занявшего второе место соотечественника Евгения Орла на 21 очко, после чего заметно оторвался от конкурентов.

Так как в беге со стрельбой Полозков также чувствует себя уверенно, то в этой ситуации между ним и победой могла встать только доставшаяся ему в конкуре лошадь по кличке Оливия. Но Оливия Илью не подвела, и на беговую трассу Полозков отправился первым, с 23-секундным отрывом от Орла. Лучшие из россиян, Кирилл Беляков и Егор Громадский, уступали лидеру 36 и 47 секунд соответственно.

Лазер-ран интригу в борьбе за первое место не вернул. Илья Полозков очень уверенно и спокойно провел и бег, и стрельбу и первым пересек финишный створ. А вот на второе место с четвертого вышел Егор Громадский. Евгений Орёл удержал за собой третью позицию. А призерами чемпионата России стали Павел Бабанин и Данил Калимуллин.

КУБОК ПАВЛА ЛЕДНЁВА

Второй этап

Москва, 27 мая 2022

Мужчины. Личное первенство

1. Илья Полозков (Беларусь) – 1516.
2. Егор Громадский (Московская область) – 1476.
3. Евгений Орёл (Беларусь) – 1461.

Рисунок 4.13 – Страница «Новости»

Страница «О спорте»

На странице «О спорте» представлена общая информация о данном виде спорта, на котором специализируется данное веб-приложение (рисунок 4.14).

Пятиборье



Современное пятиборье – уникальное соревнование, в котором чемпионом становится атлет, лучше других выступивший в пяти различных видах спорта: фехтовании, плавании, верховой езде (конкуре), беге и стрельбе. Последние две дисциплины объединены в один комбинированный вид (комбайн).

Плавание

В плавании спортсмены соревнуются на дистанции 200 метров вольным стилем (в эстафетах – 2 x 100 метров). За результат 2.30,00 спортсмен (или команда) получает 250 очков, каждые 0,5 секунды лучше ли хуже этого результата увеличивают или уменьшают его баллы на 1 очко. Например, результат 2.20,00 (или даже 2.20,49) принесет спортсмену (команде) 270 очков, результат 2.20,50 (вплоть до 2.20,99 включительно) – 269 очков, а результат 2.33,00 – 244 очка.

Заплывы формируются в соответствии с результатами спортсменов именно в этой дисциплине, плавании. Все сильнейшие пловцы среди участников турнира стартуют в последнем заплыве.

Фехтование

Пятиборцы выступают в фехтовании на шпагах. Соревнования по фехтованию проводятся по круговой системе – каждый фехтовальщик встречается с каждым. Бой проводится до одного укола. Чтобы набрать 250 очков, спортсмен должен выиграть 70% поединков. В соревнованиях 36 спортсменов каждый атлет проводит 35 боев, и для того, чтобы получить 250 очков, он должен выиграть 25 из них. Каждая победа выше или ниже этого результата оценивается в 6 очков. То есть спортсмен, одержавший 30 побед, наберет 280 очков, а тот, кто выигрывает только 15 поединков, получит 190 баллов.

Рисунок 4.14 – Страница «О спорте»

Страница «Документы»

На странице «Документы» отображены важные документы для всех ролей пользователей веб-приложения (Тренер, Организатор, Спортсмен). Все документы можно скачать в формате pdf. Страница «Документы» изображена на рисунке 4.15.

Документы

[Положение о межрегиональных и всероссийских официальных спортивных соревнованиях](#)

[Положение о статусе спортсменов](#)

[Положение о судейской коллегии](#)

[Положение о тренерском совете](#)

[Правила по современному пятиборью](#)

[Техническое задание организаторам всероссийских соревнований](#)

[Устав Федерации современного пятиборья России](#)

Рисунок 4.15 – Страница «Документы»

Оформление веб-приложения является адаптивным к различным устройствам вывода информации. На рисунках 4.16 представлено изображение мобильных версий в браузере Google Chrome с использованием инструментов для веб-разработки по тестированию отображения на различных устройствах.

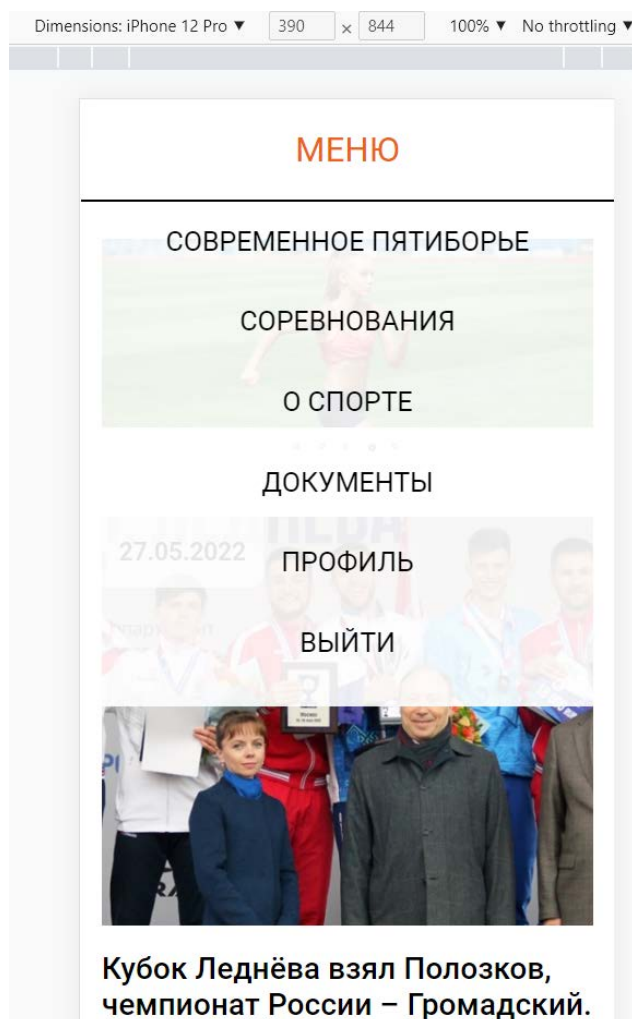
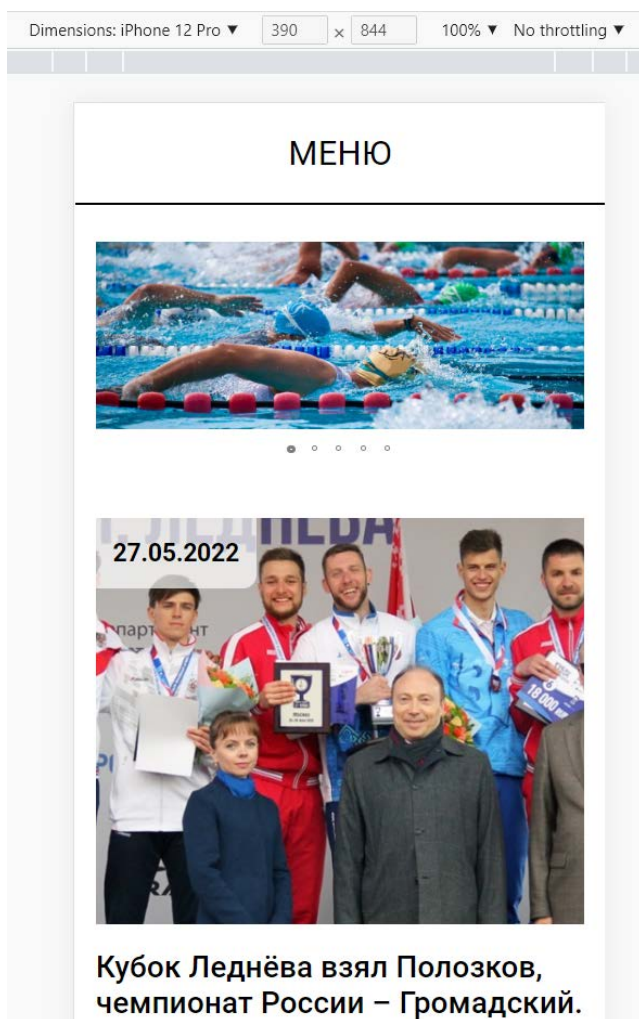


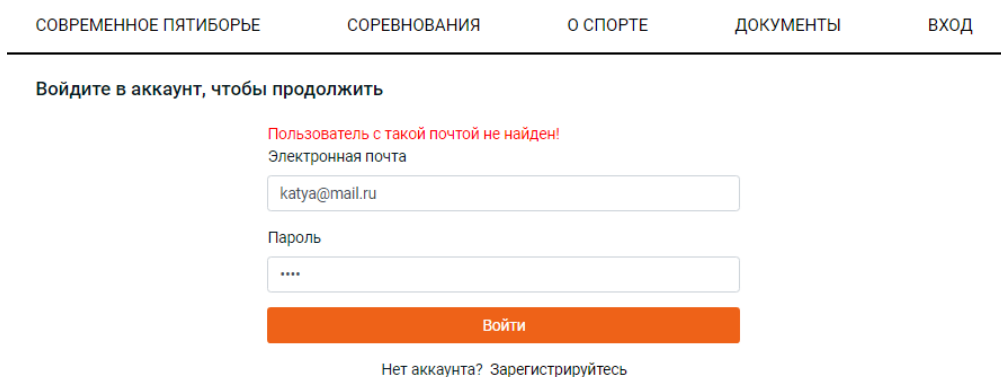
Рисунок 4.16 – Главная страница и меню веб-приложения на iPhone 12 Pro

5. ТЕСТИРОВАНИЕ

Разрабатываемый программно-аппаратный комплекс в дальнейшем будет дорабатываться, расширяя функционал для проведения как соревновательного, так и тренировочного процесса, включая сбор/анализ/обработку результатов выполнения спортсменом отдельных видов многоборья. На данный момент было проведено только альфа-тестирование программной части комплекса.

Тестирование регистрации и авторизации пользователя

Если ранее незарегистрированный пользователь попытается авторизоваться, то увидит следующую надпись (рисунок 5.1).



СОВРЕМЕННОЕ ПЯТИБОРЬЕ СОРЕВНОВАНИЯ О СПОРТЕ ДОКУМЕНТЫ ВХОД

Войдите в аккаунт, чтобы продолжить

Пользователь с такой почтой не найден!

Электронная почта

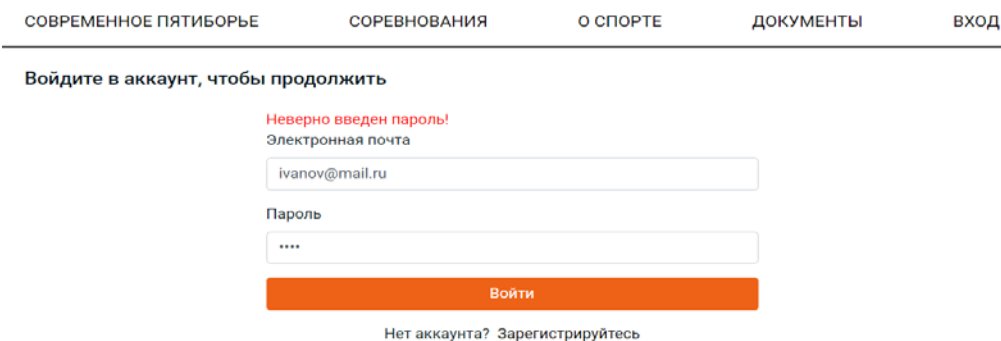
Пароль

Войти

[Нет аккаунта? Зарегистрируйтесь](#)

Рисунок 5.1 – Авторизация незарегистрированного пользователя

При вводе неправильного пароля в процессе авторизации пользователь увидит следующую надпись (рисунок 5.2).



СОВРЕМЕННОЕ ПЯТИБОРЬЕ СОРЕВНОВАНИЯ О СПОРТЕ ДОКУМЕНТЫ ВХОД

Войдите в аккаунт, чтобы продолжить

Неверно введен пароль!

Электронная почта

Пароль

Войти

[Нет аккаунта? Зарегистрируйтесь](#)

Рисунок 5.2 – Неверный пароль пользователя


При прохождении регистрации пользователь не может оставить незаполненными обязательные поля, отмеченные в форме символом *, иначе они будут выделены и отмечены специальной надписью (рисунок 5.3).

СОВРЕМЕННОЕ ПЯТИБОРЬЕ СОРЕВНОВАНИЯ О СПОРТЕ ДОКУМЕНТЫ ВХОД

Регистрация

Фамилия*
Фомин

Имя*

Отчество  Заполните это поле.
Сергеевич

Дата рождения*
02.12.1985

Город*
Челябинск ▼

Пол*
Мужской ▼

Электронная почта*
fomink@mail.ru

Телефон*
+7 999 965 68 52

Пароль*
....

Повторите пароль еще раз*
....

Моя роль*
Организатор ▼

Категория*
Первая ▼

Рисунок 5.3 – Незаполненное обязательное поле при регистрации

Пользователь не может повторно зарегистрироваться с ранее используемой электронной почтой. Пример регистрации с использованием ранее зарегистрированной электронной почты представлен на рисунке 5.4.

СОВРЕМЕННОЕ ПЯТИБОРЬЕ СОРЕВНОВАНИЯ О СПОРТЕ ДОКУМЕНТЫ ВХОД

Регистрация

Пользователь с такой почтой уже существует!

Фамилия*

Имя*

Отчество

Дата рождения*

Город*

Пол*

Электронная почта*

Телефон*

Пароль*

Повторите пароль еще раз*

Моя роль*

Разряд*

[Зарегистрироваться](#)

[Уже зарегистрированы? Войти](#)

Рисунок 5.4 – Регистрация с использованием ранее зарегистрированной
ПОЧТЫ

Тестирование создания и удаления соревнований

Для того чтобы удалить выбранное соревнование необходимо нажать на красный символ «×», изображенный справа от информации о соревновании. После этого появится подтверждающее окно. Экранная форма удаления соревнования изображена на рисунке 5.5.

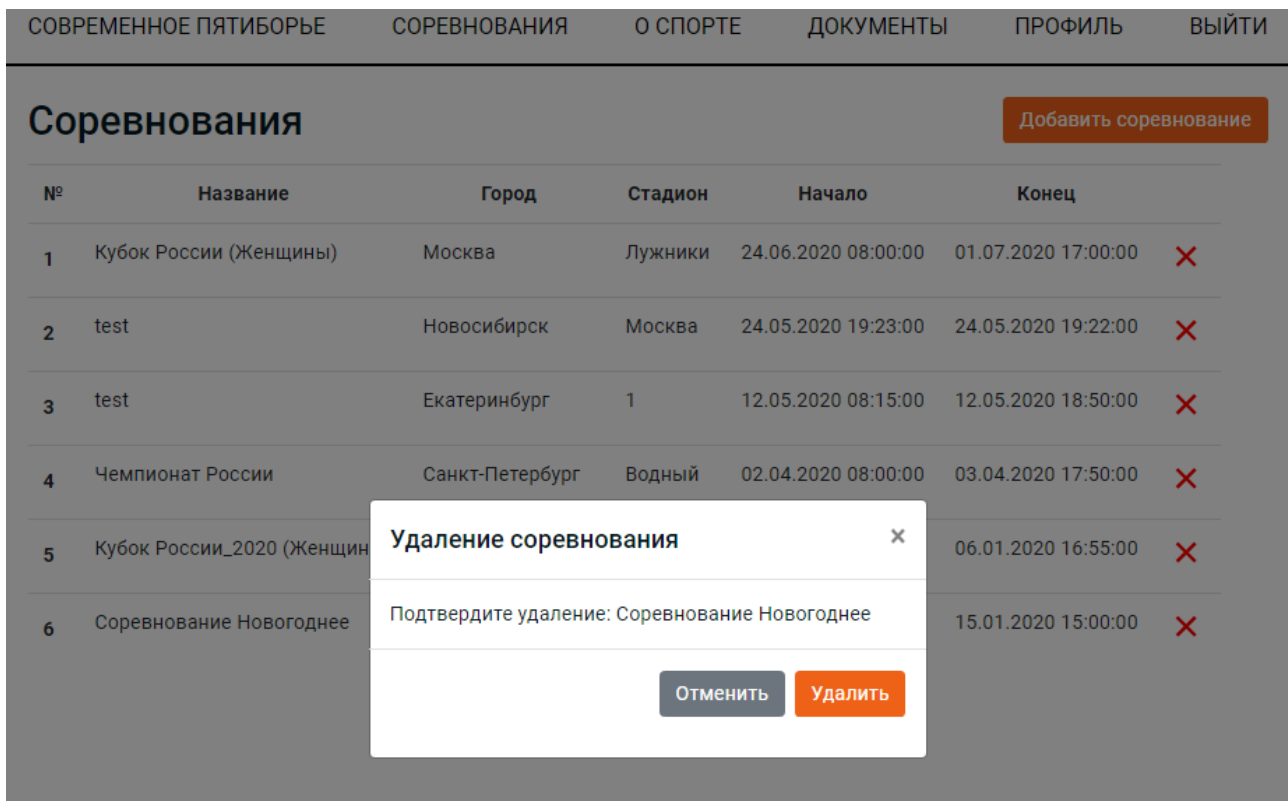


Рисунок 5.5 – Удаление соревнований

Тестирование добавления и удаления спортсмена из соревнования

Для того чтобы удалить конкретного спортсмена из соревнования необходимо нажать на символ «×», изображенный справа от информации о спортсмене. После этого появится подтверждающее окно (рисунок 5.6).

№	ФИО	Город	Год рождения	Фехтование	Плавание	Конкур	Комбайн	Общие баллы	
1	Баташова Ульяна Евгеньевна	Москва	1994	242 (6)	280 (02:15)	292	1273 (00:27)	2087	✗
2	Кошкина Александра Ивановна	Самара	2003	234 (5)	186 (03:02)	300	1206 (01:34)	1926	✗
3	Чистякова Анастасия Леонидовна					272	1283 (00:17)	1805	✗
4	Губайдуллина Гульназ Рифкатовна					100	1195 (01:45)	1801	✗
5	Фральцова Ксения Игоревна					282	1177 (02:03)	1685	✗
6	Тебекина Людмила Дмитриевна	Нижний Новгород	1988	258 (8)	0	292	1059 (04:01)	1609	✗

Удаление соревнования ✕

Подтвердите удаление Баташова Ульяна Евгеньевна из данного соревнования

Отменить
Удалить

Рисунок 5.6 – Удаление спортсмена из соревнования

Тестирование на проведении конкретного соревнования

Для проверки корректности обработки данных и их перевода в балльно-рейтинговую систему проведено тестирование данного комплекса на основе результатов одного из ранее проведенных соревнований.

Для начала создадим новое соревнование с названием «Первенство России (Женщины) 2022 г.» (рисунок 5.6).

Добавление соревнования

Название *

Тип соревнований

Город*

Стадион*

Начало*

Конец*

Судьи*

- Иванов Петр Иванович
- Непалов Игорь Алексеевич
- Терехова Светлана Семеновна
- Тестовой Сергей Алексеевич
- Судья Для Тренировок

Рисунок 5.6 – Создание соревнования Первенство России (Женщины)2022 г.

После чего необходимо добавить всех участников (спортсменов) данного старта из ранее зарегистрированных спортсменов (рисунок 5.7).

Добавление спортсменов

- 1) Авдеева Алена Александровна (г. Нижний Новгород)
- 2) Баташова Ульяна Евгеньевна (г. Москва)
- 3) Буряк Анна Анатольевна (г. Москва)
- 4) Губайдуллина Гульназ Рифкатовна (г. Москва)
- 5) Епифанова Анна Сергеевна (г. Москва)
- 6) Камылев Игорь Сергеевич (г. Нижний Новгород)
- 7) Кошкина Александра Ивановна (г. Самара)
- 8) Орлова Ирина Сергеевна (г. Новосибирск)
- 9) Сергеева Юлия Сергеевна (г. Челябинск)
- 10) Сорокин Петр Олегович (г. Екатеринбург)
- 11) Специально Для Тренировок (г. Москва)
- 12) Тебекина Людмила Дмитриевна (г. Нижний Новгород)
- 13) Федеров Сергей Семенович (г. Челябинск)
- 14) Фральцова Ксения Игоревна (г. Москва)
- 15) Хомякова Татьяна Федоровна (г. Уфа)
- 16) Хураськина Екатерина Олеговна (г. Москва)
- 17) Чистякова Анастасия Леонидовна (г. Москва)

Добавить

Рисунок 5.7 – Добавление спортсменов в Первенство России (Женщины) 2022 г.

На рисунке 5.8 представлено внесение результатов для спортсменов по каждому виду многоборья.

Введите данные по каждому спортсмену

Авдеева Алена Александровна (г. Нижний Новгород)

Фехтование

Кол-во выигранных боев: 10



0 12

Плавание (в минутах)

02:31



Конкур

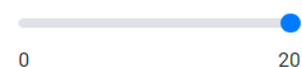
293

Комбайн (в минутах)

11:39



Попаданий в мешень: 20

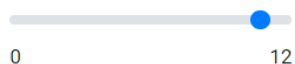


0 20

Баташова Ульяна Евгеньевна (г. Москва)

Фехтование

Кол-во выигранных боев: 11



0 12

Плавание (в минутах)

02:11



Конкур

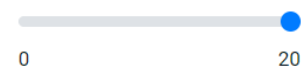
300

Комбайн (в минутах)

11:02



Попаданий в мешень: 20

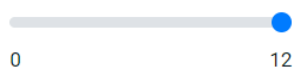


0 20

Буряк Анна Анатольевна (г. Москва)

Фехтование

Кол-во выигранных боев: 12



0 12

Плавание (в минутах)

02:17



Конкур

279

Комбайн (в минутах)

11:15



Попаданий в мешень: 19



0 20

Рисунок 5.8 – Ввод результатов по каждому виду многоборья

На рисунке 5.9 изображен итоговый протокол Первенства России (Женщины) 2022 г.

Первенство России (Женщины) 2022 г.

№	ФИО	Город	Год рождения	Фехтование	Плавание	Конкур	Комбайн	Общие баллы	
1	Баташова Ульяна Евгеньевна	Москва	1994	274 (11)	288 (02:11)	300	638 (11:02)	1500	✗
2	Буряк Анна Анатольевна	Москва	1991	282 (12)	276 (02:17)	279	625 (11:15)	1462	✗
3	Губайдуллина Гульназ Рифкатовна	Москва	1992	242 (7)	302 (02:04)	258	630 (11:10)	1432	✗
4	Чистякова Анастасия Леонидовна	Москва	1997	266 (10)	244 (02:33)	300	620 (11:20)	1430	✗
5	Авдеева Алена Александровна	Нижний Новгород	1999	266 (10)	248 (02:31)	293	601 (11:39)	1408	✗
6	Фральцова Ксения Игоревна	Москва	1998	218 (4)	266 (02:22)	300	607 (11:33)	1391	✗
7	Хураскина Екатерина Олеговна	Москва	1989	258 (9)	232 (02:39)	297	601 (11:39)	1388	✗
8	Тебекина Людмила Дмитриевна	Нижний Новгород	1988	202 (2)	252 (02:29)	266	574 (12:06)	1294	✗
9	Сергеева Юлия Сергеевна	Челябинск	2001	242 (7)	198 (02:56)	300	504 (13:16)	1244	✗
10	Орлова Ирина Сергеевна	Новосибирск	1995	218 (4)	214 (02:48)	299	472 (13:48)	1203	✗
11	Кошкина Александра Ивановна	Самара	2003	202 (2)	260 (02:25)	0	564 (12:16)	1026	✗
12	Епифанова Анна Сергеевна	Москва	2000	218 (4)	238 (02:36)	0	546 (12:34)	1002	✗
13	Хомякова Татьяна Федоровна	Уфа	2002	194 (1)	228 (02:41)	0	549 (12:31)	971	✗

Рисунок 5.9 – Итоговый протокол Первенства России (Женщины) 2022 г.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был разработан программно-аппаратный комплекс для организации соревнований и тренировок по современному пятиборью.

Для достижения поставленной цели были решены следующие задачи:

1. Выполнен анализ предметной области.
2. Проведен анализ существующих аналогов, выявлены их преимущества и недостатки.
3. Сформированы требования к разрабатываемому комплексу.
4. Разработана структура программно-аппаратного комплекса.
5. Спроектирована аппаратная часть комплекса:
 - разработана функциональная схема модуля сопряжения;
 - сформулированы требования и подобраны элементы;
 - разработана принципиальная схема модуля сопряжения.
6. Разработана программная часть комплекса:
 - выбраны среда разработки и средства реализации;
 - разработана архитектура веб-приложения;
 - спроектирована структура базы данных;
 - реализован функционал веб-приложения.
7. Произведено тестирование работы программно-аппаратного комплекса.

В настоящее время работа над комплексом продолжается с целью расширения функциональных возможностей. Основной функционал комплекса готов к опытной эксплуатации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. История пятиборья. – Режим доступа: <https://www.pentathlon.by/sovremennoe-pyatibore/>. Дата обращения: 10.01.2022.
2. МБУ «Спортивная школа Конноспортивный комплекс Рифей им. П.М.Латышева» г. Челябинска – Режим доступа: <http://ksk-rifej.ru/Pentathlon/>. Дата обращения: 15.01.2022.
3. Мишени S-Box PENTASHOT. – Режим доступа: <http://pentashot.eu/Public/Files/Article/manual-rus.pdf>. Дата обращения: 01.03.2022.
4. Программное обеспечение SportGeek. – Режим доступа: <http://www.sportgeek.uk/>. Дата обращения: 01.03.2022.
5. Критерии выбора СУБД. – Режим доступа: <https://studizba.com/lectures/10-informatika-i-programmirovanie/362-proektirovanie-baz-dannyh-na-sql/4880-10-kriterii-vybora-subd.html>. Дата обращения: 11.05.2022.
6. Сравнение систем управления базами данных. – Режим доступа: <https://devacademy.ru/article/sqlite-vs-mysql-vs-postgresql/>. Дата обращения: 11.04.2022.
7. Сравнение веб-фреймворков React и Angular. – Режим доступа: <https://artjoker.ua/ru/blog/chto-luchshe-react-ili-angular/>. Дата обращения: 12.02.2022.
8. MongoDB. – <https://metanit.com/nosql/mongodb/1.1.php>. Дата обращения: 12.02.2022.
9. NodeJS. – <https://senior.ua/articles/pochemu-node-js-osobennosti-i-preimuschestva>. Дата обращения: 22.02.2022.

10. PHP. – <https://stfalcon.com/ru/blog/post/PHP-advantages-for-business>. Дата обращения: 22.02.2022.
11. Bootstrap. – https://depix.ru/articles/bootstrap_preimuschestva_i_nedostatki. Дата обращения: 28.02.2022.
12. Список сертифицированного стрелкового оборудования. – [https://www.uipmworld.org/uipm-homologated-equipment](https://www UIPMWorld.org/UIPM-homologated-equipment). Дата обращения: 15.04.2022.
13. Программатор для микроконтроллера. – <https://www.phyton.ru/programmers/programmer-for-beginners>. Дата обращения: 25.04.2022.
14. Кварцевый резонатор. – <https://radiostorage.net/1500-stabilizaciya-chastoty-mikrokontrollera-kvarcevym-rezonatorom.html>. Дата обращения: 25.04.2022.
15. Wi-Fi модуль ESP8266. – <https://arduinomaster.ru/platy-arduino/arduino-esp8266/>. Дата обращения: 22.04.2022.
16. ATmega8L. Datasheet. – Atmel, 2013. – 330 с.
17. SMD Chip led lamp. Datasheet. –Kingbright, 2001. – 4с.
18. Предохранители. – <https://electrono.ru/elektrotexnicheskaya-apparatura/predoxraniteli>. Дата обращения: 22.04.2022.
19. Контроллер заряда MCP73831. – <https://4donline.ihs.com/images/VipMasterIC/IC/MCHP/MCHPS05673/MCHPS05673-1.pdf?hkey=6D3A4C79FDBF58556ACFDE234799DDF0>. Дата обращения: 06.04.2022.

20. Стабилизатор напряжения. – https://www.td-m.ru/resheniya/tipy_i_printsip_raboty_stabilizatora_napryazheniya/. Дата обращения: 27.04.2022.

21. Языки программирования для веб-разработки. – <https://webshake.ru/post/yazyki-programmirovaniya-dlya-veb-razrabotki>. Дата обращения: 12.05.2022.

22. Электрофиксатор уколов Favero. – <https://pentashot.ru/mnogofunkcionalniy-elektrofixsator-favero-fa-07-1> Дата обращения: 09.02.2022.

23. Tailwind CSS. – <https://timeweb.com/ru/community/articles/chto-takoe-tailwindcss-zachem-nuzhen-i-chem-horosh> Дата обращения: 17.03.2022.

ПРИЛОЖЕНИЕ А

Листинг А.1 - Исходный код Register

```
<?php require "config/config.php";
// Запрос городов
$cities = getConnectionDB()->query("SELECT * FROM cities ORDER BY name")-
>fetchAll(PDO::FETCH_OBJ);
// Запрос ролей
$roles = getConnectionDB()->query("SELECT * FROM roles ORDER BY id DESC")-
>fetchAll(PDO::FETCH_OBJ);

$data = $_POST;
if (isset($data['register'])) {
    $errors = array();
    if ($data['password'] != $data['pass_check']) {
        $errors[] = 'Повторный пароль введен неверно!';
    }

    if (checkEmailAvailability($data['email']) > 0) {
        $errors[] = 'Пользователь с такой почтой уже существует!';
    }

    if (empty($errors)) {
        insertUser($data); // Внесение данных в бд
        header("Location: /profile.php");
        die();
    } else {
        $errorText = "<div style=\"color: red\">" . array_shift($errors) . "</div>";
    }
}

require "header.php";
?>

<section>
    <div class="container">
        <div class="row">
            <div class="col-12">
                <h3>Регистрация</h3>
            </div>
        </div>
        <div class="row">
            <div class="col-12 offset-md-2 col-md-8">
                <div class="content-wrapper" style="max-width: 560px; margin: 0
auto;">
                    <form method="post" class="register">
                        <div class="row">
                            <div class="col-12 test">
                                <?php if (!empty($errorText)) echo $errorText; ?>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</section>
```

```

        <label for="surname">Фамилия*</label>
        <input required type="text" class="form-control"
id="surname" name="surname"
            value="<?= @$data['surname']; ?>">
    </div>
    <div class="form-group">
        <label for="name">Имя*</label>
        <input required type="text" class="form-control"
id="name" name="name"
            value="<?= @$data['name']; ?>">
    </div>
    <div class="form-group">
        <label for="patronymic">Отчество</label>
        <input type="text" class="form-control"
id="patronymic" name="patronymic"
            value="<?= @$data['patronymic']; ?>">
    </div>

    <div class="form-group">
        <label for="date">Дата рождения*</label>
        <input required type="date" class="form-control"
id="date" name="birth_date"
            value="<?= @$data['birth_date']; ?>">
    </div>

    <div class="form-group">
        <label for="city">Город*</label>
        <select required class="form-control" id="city"
name="city">
            <?php for ($i = 0; $i < count($cities); $i++): ?>
                <option value="<?= $cities[$i]->id ?>"><?=
$cities[$i]->name ?></option>
            <?php endfor; ?>
        </select>
    </div>

    <div class="form-group">
        <label for="gender">Пол*</label>
        <select required class="form-control"
id="gender" name="gender">
            <option>Мужской</option>
            <option>Женский</option>
        </select>
    </div>

    <div class="form-group">
        <label for="email">Электронная почта*</label>
        <input required type="email" name="email"
id="email" class="form-control"
            value="<?= @$data['email']; ?>">
    </div>
    <div class="form-group">
        <label for="phone">Телефон*</label>

```

```

        <input required type="text" class="form-control"
id="phone" name="phone"
        value="<?= @$data['phone']; ?>"
    </div>

    <div class="form-group">
        <label for="password">Пароль*</label>
        <input required type="password" name="password"
id="password" class="form-control"
        value="<?= @$data['password']; ?>"
    </div>
    <div class="form-group">
        <label for="pass_check">Повторите пароль еще
раз*</label>
        <input
            required
            type="password"
name="pass_check" id="pass_check" class="form-control"
        value="<?= @$data['pass_check']; ?>"
    </div>

    <div class="form-group">
        <label for="role">Моя роль*</label>
        <select required class="form-control" role="
id="role" name="role">
            <?php for ($i = 0; $i < count($roles); $i++): ?>
                <option value="<?= $roles[$i]->id ?>"><?=
$roles[$i]->name ?></option>
            <?php endfor; ?>
        </select>
    </div>

    <div class="form-group div-category">
        <label
            for="category"
            class="name-
category">Категория*</label>
        <select required class="form-control" category="
id="category" name="category">
            </select>
    </div>

    <div class="form-group div-experience">
    </div>

    <div class="form-group div-type">
    </div>

    <button type="submit" class="btn button btn-100"
name="register">Зарегистрироваться</button>
    </div>
</div>

<div class="row">
    <div class="col-12 text-center">
        <p
            class="d-inline-block">Уже
зарегистрированы?&nbsp;</p>
    </div>
</div>

```



```

        <a href="/login.php ">Войти</a>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
</section>
<?php require "footer.php"; ?>

```

Листинг А.2 - Исходный код Profile

```

<?php require "config/config.php";
// Человек пришел после регистрации
if (!empty($_SESSION['user_email'])) {
    $_SESSION['user'] = getUserByEmail($_SESSION['user_email']);
    $user = $_SESSION['user'];
    unset($_SESSION['user_email']);
}

// Защита от гостей
if (empty($_SESSION['user'])) {
    header("Location: /login.php");
    die();
}

if (!empty($_SESSION['user'])) {
    $_SESSION['user'] = getUserByEmail($_SESSION['user']->email);
    $user = $_SESSION['user'];
}
require "header.php";
// Обращение к бд
$comp = getAllCompetitionsByStart();
?>

<section>
    <div class="container">
        <div class="row">
            <div class="col">
                <h1>Личный кабинет</h1>
            </div>

            <div class="col">
                <a href="/profile_change.php" class="btn button float-right">Изменить
профиль</a>
            </div>
        </div>

        <?php if (!empty($user)): ?>
            <div class="row">
                <div class="col-12 col-lg-4">
                    <div class="profile">
                        <div>Фамилия: <?= $user->surname; ?></div>

```

```

<div>Имя: <?= $user->name; ?></div>

<?php if (!empty($user->patronymic)): ?>
    <div>Отчество: <?= $user->patronymic; ?></div>
<?php endif; ?>

<div>Дата рождения: <?= date("d.m.Y", strtotime($user-
>birth_date)); ?></div>

<div>Пол: <?= $user->gender; ?></div>
<div>Телефон: <?= $user->phone; ?></div>
<div>Почта: <?= $user->email; ?></div>
<div>Город: <?= getCity($user->city_id); ?></div>
<div>Роль: <?= getUserRole($user->role_id); ?></div>
<?php if($user->role_id == 3): ?>
<div>Рост: <?= $user->growth; ?> см</div>
<div>Вес: <?= $user->weight; ?> кг</div>
<div>Обхват груди: <?= $user->bust; ?> см</div>
<div>Обхват талии: <?= $user->waist; ?> см</div>
<div>Обхват бедер: <?= $user->hips; ?> см</div>
<div>ИМТ: <?= imt($user->growth, $user->weight); ?></div>
<?php endif; ?>
<?php if ($user->role_id == 1 || $user->role_id == 2): ?>
    <div>Категория: <?= $user->category; ?></div>
<?php elseif ($user->role_id == 3): ?>
    <div>Разряд: <?= $user->category; ?></div>
<?php endif; ?>

<?php if ($user->role_id == 2): ?>
    <div>Стаж: <?= $user->experience; ?></div>
    <div>Вид спорта: <?= $user->type; ?></div>
<?php endif; ?>
</div>

</div>
<div class="col-12 col-lg-8">
    <table class="table table-responsive table-hover pr-comp">
        <thead>
            <tr class="text-center">
                <th>Название</th>
                <th>Город</th>
                <th>Начало</th>
                <th>Тип</th>
                <?php if ($_SESSION['user']->role_id == 1): ?>
                    <th></th>
                <?php endif; ?>
            </tr>
        </thead>
        <tbody>
            <?php if(count($comp) >= 10) {$k = 10;} else {$k =
count($comp);} ?>
            <?php for ($i = 0; $i < count($comp); $i++): ?>
                <tr class="profile-comp">

```

```

?>)" class="table-row-pr">
    <td onclick="goOneCompetition(<?= $comp[$i]->id
        <?= $comp[$i]->name; ?>
    </td>
?>)" class="table-row-pr">
    <td onclick="goOneCompetition(<?= $comp[$i]->id
        <?= getCity($comp[$i]->city_id); ?>
    </td>
?>)" class="table-row-pr">
    <td onclick="goOneCompetition(<?= $comp[$i]->id
        <?= date("d.m.Y H:i:s", strtotime($comp[$i]-
>start)); ?>
    </td>
?>)" class="table-row-pr">
    <td onclick="goOneCompetition(<?= $comp[$i]->id
        <?= $comp[$i]->types; ?>
    </td>
    </tr>
</tbody>
</table>
</div>
</div>
<?php $dost = getAllCompetitionsByUser($user->id); ?>
<div class="col-12 col-lg-12 profile-res">
    <table class="table table-responsive table-hover pr-comp">
        <thead>
            <tr class="text-center">
                <th>Соревнование</th>
                <th>Фехтование</th>
                <th>Баллы за фехтование</th>
                <th>Плавание</th>
                <th>Баллы за плавание</th>
                <th>Конкур</th>
                <th>Комбайн</th>
                <th>Стрелковые попадания</th>
                <th>Баллы за комбайн</th>
                <th>Общее количество</th>
            </tr>
        </thead>
        <tbody>
            <?php for ($i = 0; $i < count($dost); $i++): ?>
                <tr class="profile-comp">
                    <td class="table-row-pr"><?php echo
getNameCompetitionById($dost[$i]->competition_id); ?></td>
                    <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->fencing; ?></td>
                    <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->score_fencing; ?></td>
                    <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->swimming; ?></td>
                    <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->score_swimming; ?></td>

```

```

        <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->jumping; ?></td>
        <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->combine; ?></td>
        <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->combine_hits; ?></td>
        <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->score_combine; ?></td>
        <td class="table-row-pr" style="text-align: center;
vertical-align: middle;"><?php echo $dost[$i]->total_score; ?></td>
    </tr>
    <?php endfor; ?>
</tbody>
</table>
</div>

    <?php endif; ?>
</div>
</section>
<?php require "footer.php"; ?>

```

Листинг А.3 - Исходный код Competitions

```

<?php
require $_SERVER['DOCUMENT_ROOT'] . "/config/config.php";

// Защита от гостей
if (empty($_SESSION['user'])) {
    header("Location: /login.php");
    die();
}

require "header.php";

// Обращение к бд
$comp = getAllCompetitionsByStart();
?>

<section>
    <div class="container">
        <div class="row">
            <div class="col">
                <h1>Соревнования</h1>
            </div>
            <?php if ($_SESSION['user']->role_id != 3): ?>
                <div class="col">
                    <a href="add/competition.php" class="btn button float-
right">Добавить соревнование</a>
                </div>
            <?php endif; ?>
        </div>

        <?php if (!empty($comp)): ?>
            <div class="row">

```

```

<div class="col-12">
    <table class="table table-responsive table-hover">
        <thead>
            <tr class="text-center">
                <th>№</th>
                <th>Название</th>
                <th>Город</th>
                <th>Стадион</th>
                <th>Начало</th>
                <th>Конец</th>
                <th>Тип</th>
                <?php if ($_SESSION['user']->role_id == 1): ?>
                    <th></th>
                <?php endif; ?>
            </tr>
        </thead>
        <tbody class="table-competitions">
            <?php for ($i = 0; $i < count($comp); $i++): ?>
                <tr>
                    <th
                        scope="row"
                        class="num"
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= $i + 1 ?>
                    </th>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= $comp[$i]->name; ?>
                    </td>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= getCity($comp[$i]->city_id); ?>
                    </td>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= $comp[$i]->stadium; ?>
                    </td>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= date("d.m.Y H:i:s", strtotime($comp[$i]-
>start)); ?>
                    </td>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= date("d.m.Y H:i:s", strtotime($comp[$i]-
>end)); ?>
                    </td>
                    <td
                        onclick="goOneCompetition(<?= $comp[$i]->id ?>)"
                        class="table-row-comp">
                        <?= $comp[$i]->types; ?>
                    </td>
                </tr>
            <?php if ($_SESSION['user']->role_id == 1): ?>

```

```

                                <td>
                                <a href="#" data-target="#model-delete"
data-toggle="modal" class="table-delete"
                                data-content="Подтвердите удаление: <?=
$comp[$i]->name ?>"
                                data-id="<?= $comp[$i]->id ?>"></a>
                                </td>
                                <?php endif; ?>
                                </tr>
                                <?php endfor; ?>
                                </tbody>
                                </table>
                                </div>
                                </div>
                                <?php else: ?>
                                <div class="row">
                                <div class="col-lg-12">Соревнований нет!</div>
                                </div>
                                <?php endif; ?>
                                </div>
                                </section>

                                <?php if ($_SESSION['user']->role_id == 1): ?>
                                <div class="modal fade" id="model-delete" tabindex="-1" role="dialog" aria-
labelledby="exampleModalCenterTitle" aria-hidden="true">
                                <div class="modal-dialog modal-dialog-centered" role="document">
                                <div class="modal-content">
                                <div class="modal-header">
                                <h5 class="modal-title" id="exampleModalLongTitle">Удаление
соревнования</h5>
                                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                                <span aria-hidden="true">&times;</span>
                                </button>
                                </div>
                                <div class="modal-body">...</div>
                                <div class="modal-footer">
                                <button type="button" class="btn btn-secondary" data-
dismiss="modal">Отменить</button>
                                <button type="button" class="btn button btn-delete-competition"
data-dismiss="modal">Удалить</button>
                                </div>
                                </div>
                                </div>
                                </div>
                                <?php endif; ?>

                                <?php require "footer.php"; ?>

```

Листинг А.4 - Исходный код Add Result

```

<?php
require $_SERVER['DOCUMENT_ROOT'] . '/config/config.php';

```

```

// Защита от незарегистрированных пользователей
if (empty($_SESSION['user']) || $_SESSION['user']->role_id == 3) {
    header("Location: /login.php");
    die();
}

$sportsmen = getSportsmenById($_SESSION['sportsmen']);
require $_SERVER['DOCUMENT_ROOT'] . '/header.php';
?>

<section>
    <div class="container">
        <div class="row">
            <div class="col-12">
                <h3>Введите данные по каждому спортсмену</h3>
            </div>
        </div>

        <form action="/config/post.php" method="POST">
            <?php foreach ($sportsmen as $user): ?>
                <div class="row result-sportsmen">
                    <div class="col-12 form-group">
                        <?=$user->surname . ' ' . $user->name . ' ' .
                            $user->patronymic . ' (г. ' . getCity($user->city_id) . ')>';
                    </div>

                    <div class="col-12 col-md-6 col-lg">
                        <div class="form-group form-group-mb">
                            <label for="fencing-<?=$user->id ?>">Фехтование</label>
                            <label> Кол-во выигранных боев: <span class="fencing
fencing-<?=$user->id ?>">0</span></label>
                            <input type="range" class="custom-range" min="0" max="<?=$
count($sportsmen) - 1; ?>" value="0"
                                name="<?=$user->id ?>[]" id="fencing-<?=$user-
>id ?>" oninput="inputFencing(<?=$user->id ?>)">
                            <div>0<span class="float-right"><?=$ count($sportsmen) -
1; ?></span></div>
                        </div>
                    </div>

                    <div class="col-12 col-md-6 col-lg">
                        <div class="form-group">
                            <label for="swimming-<?=$user->id ?>">Плавание (в
минутах)</label>
                            <input type="time" class="form-control"
                                name="<?=$user->id ?>[]" id="swimming-<?=$user-
>id ?>">
                        </div>
                    </div>

                    <div class="col-12 col-md-6 col-lg">
                        <div class="form-group">

```

```

        <label for="jumping-= $user-&gt;id ?&gt;"&gt;Конкур&lt;/label&gt;
        &lt;input type="text" class="form-control"
            name="<?= $user-&gt;id ?&gt;[]" id="jumping-<?= $user-
&gt;id ?&gt;"&gt;
    &lt;/div&gt;
&lt;/div&gt;

&lt;div class="col-12 col-md-6 col-lg"&gt;
    &lt;div class="form-group"&gt;
        &lt;label for="combine-<?= $user-&gt;id ?&gt;"&gt;Комбайн (в
минутах)&lt;/label&gt;
        &lt;input type="time" class="form-control"
placeholder="10:00"
            name="<?= $user-&gt;id ?&gt;[]" id="combine-<?= $user-
&gt;id ?&gt;"&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;label for="hits-<?= $user-&gt;id ?&gt;"&gt;Попаданий в мешень:
&lt;span class="hits hits-<?= $user-&gt;id ?&gt;"&gt;0&lt;/span&gt;&lt;/label&gt;
        &lt;input type="range" class="custom-range" min="0" max="20"
value="0"
            name="<?= $user-&gt;id ?&gt;[]" id="hits-<?= $user-&gt;id
?&gt;" oninput="inputHits(<?= $user-&gt;id ?&gt;)"&gt;
        &lt;div&gt;0&lt;span class="float-right"&gt;20&lt;/span&gt;&lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;?php endforeach; ?&gt;

&lt;div class="row"&gt;
    &lt;div class="col text-center"&gt;
        &lt;button type="submit" name="add_result" class="btn button btn-
50"&gt;
            &lt;?php if (!empty($compId)) echo 'Изменить'; else echo
'Добавить'; ?&gt;
        &lt;/button&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/form&gt;
&lt;/div&gt;
&lt;/section&gt;

&lt;?php require $_SERVER['DOCUMENT_ROOT'] . "/footer.php"; ?&gt;
</pre

```

Листинг А.5 - Исходный код Config functions

```

<?php
// Функция подключения бд
function getConnectionDB(){
    // Данные локальной бд!
    $driver = 'mysql';
    $host = 'localhost';
    $db_name = 'db2'; // Название бд
    $db_user = 'root';

```



```

$db_pass = '';
$charset = 'utf8';
$options = [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"];
static $pdo;

if (empty($pdo)) {
    try {
        $pdo = new PDO("$driver:host=$host; dbname=$db_name; charset=$charset",
            $db_user, $db_pass, $options);
    } catch (PDOException $e){
        die ('Произошла ошибка при подключении к базе данных!');
    }
}
return $pdo;
}
function imt($growth, $weight)
{
    if ($growth == 0) return 'недостаточно данных';
    return round($weight/($growth*$growth)*10000);
}
function getScoreFencing($count, $total) {
    $cnt70 = round($total * 0.7);

    if ($count > $cnt70) {
        $score = 250 + ($count - $cnt70) * 8;
    } else {
        $score = 250 + ($count - $cnt70) * 8;
    }

    return $score;
}

function getScoreSwimming($time) {
    // 2:30 минут = 150 секунд = 250 баллов / 1:00 = 340 / 370
    $t = explode(':', $time);
    $m = $t[0] * 60;
    $s = $t[1];
    $totalTime = $m + $s;
    $points = abs($totalTime - 150);

    if ($totalTime - 150 > 0) {
        $score = 250 - $points * 2;
    } else {
        $score = 250 + $points * 2;
    }

    return $score;
}

function getScoreCombine($time) {
    // 10:00 минут = 600 секунд = 700 баллов
    $t = explode(':', $time);

```

```

    $m = $t[0] * 60;
    $s = $t[1];
    $totalTime = $m + $s;
    $points = abs($totalTime - 600);

    if ($totalTime - 600 > 0) {
        $score = 700 - $points;
    } else {
        $score = 700 + $points;
    }

    return $score;
}

// SELECT //
// Проверка существования той же почты
function checkEmailAvailability($email) {
    $stmt = getConnectionDB()->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->execute([$email]);
    $stmt->fetchAll(PDO::FETCH_OBJ);
    return $stmt->rowCount();
}

// Пользователь по id
function getUserById($id) {
    $sql = "SELECT id, surname, name, patronymic, city_id FROM users WHERE id = ?";
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([$id]);
    return $stmt->fetch(PDO::FETCH_OBJ);
}

// Пользователь по почте
function getUserByEmail($email) {
    $sql = "SELECT * FROM users WHERE email = ?";
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([$email]);
    $user = $stmt->fetch(PDO::FETCH_OBJ);

    if ($user->role_id == 2) { // Поля для тренеров
        $sql = "SELECT sport_type, experience FROM user_coach WHERE user_id = ?";
        $stmt = getConnectionDB()->prepare($sql);
        $stmt->execute([$user->id]);
        $coachFields = $stmt->fetch(PDO::FETCH_OBJ);
        $user->type = $coachFields->sport_type;
        $user->experience = $coachFields->experience;
    }

    return $user;
}

// Получить город пользователя
function getCity($city_id) {
    $sql = "SELECT * FROM cities WHERE id = ?";

```

```

$stmt = getConnectionDB()->prepare($sql);
$stmt->execute([$city_id]);
return $stmt->fetch(PDO::FETCH_OBJ)->name;
}

// Получить роль пользователя
function getUserRole($role_id) {
    $sql = "SELECT * FROM roles WHERE id = ?";
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([$role_id]);
    return $stmt->fetch(PDO::FETCH_OBJ)->name;
}

// Получить всех спортсменов по массиву айдишников
function getSportmenById($arrId) {
    $sql = 'SELECT id, surname, name, patronymic, city_id FROM users WHERE ';
    for ($i = 0; $i < count($arrId); $i++) {
        if ($i === count($arrId) - 1) {
            $sql .= "id = $arrId[$i] ";
        } else {
            $sql .= "id = $arrId[$i] OR ";
        }
    }

    $sql .= 'ORDER BY surname ASC';
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([]);
    return $stmt->fetchAll(PDO::FETCH_OBJ);
}

// Получить все соревнования, сначала новые
function getAllCompetitionsByStart($params = '*') {
    $sql = "SELECT $params FROM competitions ORDER BY start DESC";
    return getConnectionDB()->query($sql)->fetchAll(PDO::FETCH_OBJ);
}

// Получить все соревнования в которых участвовал пользователь
function getAllCompetitionsByUser($id) {
    $sql = "SELECT * FROM competition_sportsman WHERE user_id=$id";
    return getConnectionDB()->query($sql)->fetchAll(PDO::FETCH_OBJ);
}

// Получить одно соревнование по id
function getNameCompetitionById($id) {
    $sql = "SELECT c.name FROM competitions AS c WHERE c.id = ? ";
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([$id]);
    return $stmt->fetch(PDO::FETCH_OBJ)->name;
}

// Получить данные спортсмена по соревнованию
function getDataCompetition($id) {
    $sql = "SELECT c_s.*, u.id as user_id, u.surname, u.name, u.patronymic,
u.birth_date, u.city_id

```

```

FROM competition_sportsman as c_s, users as u
WHERE c_s.competition_id = :id AND c_s.user_id = u.id ORDER BY c_s.total_score
DESC";

$stmt = getConnectionDB()->prepare($sql);
$stmt->execute(['id' => $id]);
return $stmt->fetchAll(PDO::FETCH_OBJ);
}

// Получить новости
function getNews() {
    $sql = "SELECT * FROM news ORDER BY pubdate DESC";
    return getConnectionDB()->query($sql)->fetchAll(PDO::FETCH_OBJ);
}

// Получить новость по id
function getNewsById($id) {
    $sql = 'SELECT * FROM news WHERE id = ?';
    $stmt = getConnectionDB()->prepare($sql);
    $stmt->execute([$id]);
    return $stmt->fetch(PDO::FETCH_OBJ);
}

// INSERT //
// Вставка пользователя в бд
function insertUser($data) {
    // Проверка введенных данных
    $params = [
        'surname' => htmlspecialchars(trim($data['surname'])),
        'name' => htmlspecialchars(trim($data['name'])),
        'patronymic' => htmlspecialchars(trim($data['patronymic'])),
        'birth_date' => $data['birth_date'],
        'city' => htmlspecialchars(trim($data['city'])),
        'gender' => htmlspecialchars(trim($data['gender'])),
        'email' => htmlspecialchars(trim($data['email'])),
        'phone' => htmlspecialchars(trim($data['phone'])),
        'pas' => password_hash($data['password'], PASSWORD_DEFAULT),
        'role' => htmlspecialchars(trim($data['role'])),
        'category' => htmlspecialchars(trim($data['category'])),
        'weight' => $data['weight'],
        'growth' => $data['growth'],
        'bust' => $data['bust'],
        'hips' => $data['hips'],
        'waist' => $data['waist'],
        'imt' => $data['imt'],
    ];
    $sql = "INSERT INTO users (surname, name, patronymic, birth_date,
        city_id, gender, email, phone, password, role_id, category, weight, bust,
hips, waist, growth, imt)
        VALUES (:surname, :name, :patronymic, :birth_date,
            :city, :gender, :email, :phone, :pas, :role, :category, :weight, :bust,
:waist, :hips, :growth, :imt)";
    getConnectionDB()->prepare($sql)->execute($params);

    if ($data['role'] == 2) { // Если выбрана роль Тренера

```

```

        $coach = "INSERT INTO user_coach (sport_type, experience, user_id)
        VALUES (?, ?, LAST_INSERT_ID())";
        getConnectionDB()->prepare($coach)->execute([$data['type'],
$data['experience']]);
    }

    $_SESSION['user_email'] = $data['email'];
}

// Вставка соревнования в бд
function insertCompetition($data) {
    $params = [
        'name' => htmlspecialchars(trim($data['name'])),
        'stadium' => htmlspecialchars(trim($data['stadium'])),
        'start' => $data['start'],
        'end' => $data['end'],
        'city' => htmlspecialchars(trim($data['city'])),
        'types' => $data['types'],
    ];
    $sql = "INSERT INTO competitions (name, stadium, start, end, city_id, types)
    VALUES (:name, :stadium, :start, :end, :city, :types)";
    getConnectionDB()->prepare($sql)->execute($params);

    $sqlLastId = "SELECT * FROM competitions WHERE id = LAST_INSERT_ID()";
    $id = getConnectionDB()->query($sqlLastId)->fetch(PDO::FETCH_OBJ)->id;

    // Вставка категорий вставленного товара в бд
    $sqlInsert = "INSERT INTO competition_referee (user_id, competition_id) VALUES ";
    for ($i = 0; $i < count($data['referees']); $i++) {
        if ($i === count($data['referees']) - 1) {
            $sqlInsert .= "(" . $data['referees'][$i] . ", $id)";
        } else {
            $sqlInsert .= "(" . $data['referees'][$i] . ", $id), ";
        }
    }
    getConnectionDB()->prepare($sqlInsert)->execute([]);

    return $id;
}

// Вставка в результатов спортсменов в бд
function insertResultCompetition($data) {
    $compId = $_SESSION['comp_id']; // id соревнования

    $sqlInsert = "INSERT INTO competition_sportsman (competition_id, user_id,
    fencing, score_fencing, swimming, score_swimming, jumping, combine,
    score_combine, combine_hits, total_score) VALUES
    ";

    $sportsmen = $_SESSION['sportsmen'];

    // Проход по каждому спортсмену
    for ($k = 0; $k < count($sportsmen); $k++) { // Массив id спортсменов

```

```

$id = $sportsmen[$k];
$user = $data[$id]; // Массив результатов спорсмена по id
$sqlInsert .= "($compId, $id, "; // начало values у каждого спортсмена

// Проход по каждому виду результата у спортсмена
$fencing = null;
$swimming = null;
$combine = null;
$jumping = null;
for ($i = 0; $i < count($user); $i++) {
    // Запись результатов по каждому виду
    if (empty($user[$i])) {
        $sqlInsert .= "NULL, ";
    } else {
        $sqlInsert .= "'$user[$i]', ";
    }

    // Расчет баллов по фехтованию
    if ($i == 0) {
        if (!empty($user[$i])) {
            // Расчет баллов по фехтованию
            $fencing = getScoreFencing($user[$i], count($sportsmen) - 1);
            $sqlInsert .= "$fencing, ";
        } else {
            $sqlInsert .= "NULL, ";
        }
    }

    // Расчет баллов по плаванию
    if ($i == 1) {
        if (!empty($user[$i])) {
            // Расчет баллов по плаванию
            $swimming = getScoreSwimming($user[$i]);
            $sqlInsert .= "$swimming, ";
        } else {
            $sqlInsert .= "NULL, ";
        }
    }

    // Конкур
    if ($i == 2 && !empty($user[$i])) {
        $jumping = $user[$i];
    }

    // Расчет баллов по камбайну и запись общих баллов
    if ($i == 3) {
        if (!empty($user[$i])) {
            // Расчет баллов по камбайну
            $combine = getScoreCombine($user[$i]);
            $sqlInsert .= "$combine, ";
        } else {
            $sqlInsert .= "NULL, ";
        }
    }
}

```

```

    }

    // Последнее значение у каждого спортсмена
    if ($i == 4) {
        $totalScore = $fencing + $swimming + $jumping + $combine;
        if ($k === (count($sportsmen) - 1)) { // Проверка последний ли это
спортсмен
            $sqlInsert .= ($totalScore) . ")";
        } else {
            $sqlInsert .= ($totalScore) . "), ";
        }
    }
}

getConnectionDB()->prepare($sqlInsert)->execute([]);
}

// DELETE //
// Удаление студента из соревнования из бд
function deleteSportsman($data) {
    $params = [
        'id' => $data['id'],
        'comp_id' => $data['comp_id'],
    ];
    $sql = "DELETE FROM competition_sportsman
        WHERE competition_id = :comp_id AND user_id = :id";
    getConnectionDB()->prepare($sql)->execute($params);
}

// Удаление соревнования из бд
function deleteCompetitionById($id) {
    $sql = "DELETE FROM competitions WHERE id = ?";
    getConnectionDB()->prepare($sql)->execute([$id]);
}

// UPDATE
function updateUser($data, $id, $role) {
    $params = [
        'surname' => htmlspecialchars(trim($data['surname'])),
        'name' => htmlspecialchars(trim($data['name'])),
        'patronymic' => htmlspecialchars(trim($data['patronymic'])),
        'date' => $data['birth_date'],
        'city' => htmlspecialchars(trim($data['city'])),
        'gender' => htmlspecialchars(trim($data['gender'])),
        'email' => htmlspecialchars(trim($data['email'])),
        'phone' => htmlspecialchars(trim($data['phone'])),
        'category' => htmlspecialchars(trim($data['category'])),
        'id' => (int) $id,
        'weight' => htmlspecialchars(trim($data['weight'])),
        'bust' => htmlspecialchars(trim($data['bust'])),
        'hips' => htmlspecialchars(trim($data['hips'])),
        'waist' => htmlspecialchars(trim($data['waist'])),
    ];
}

```

```

        'growth' => htmlspecialchars(trim($data['growth'])),
        'imt' => htmlspecialchars(trim($data['imt'])),
    ];

    // Если пароль был изменен
    if (!empty($data['password'])) {
        $params['pas'] = password_hash($data['password'], PASSWORD_DEFAULT);

        $sql = "UPDATE users SET surname=:surname, name=:name,
patronymic=:patronymic, birth_date=:date,
                city_id=:city, gender=:gender, email=:email, phone=:phone,
category=:category, password=:pas, weight=:weight, bust=:bust, hips=:hips, waist=:waist,
growth=:growth, imt=:imt
                WHERE id=:id";
    }
    else {
        $sql = "UPDATE users SET surname=:surname, name=:name,
patronymic=:patronymic, birth_date=:date,
                city_id=:city, gender=:gender, email=:email, phone=:phone,
category=:category, weight=:weight, bust=:bust, hips=:hips, growth=:growth, waist=:waist,
imt=:imt
                WHERE id=:id";
    }
    getConnectionDB()->prepare($sql)->execute($params);

    // Обновление полей у Тренера
    if ($role == 2) {
        $coach = "UPDATE user_coach SET sport_type=?, experience=?
                WHERE user_id=?";
        getConnectionDB()->prepare($coach)->execute([$data['type'],
$data['experience'], $id]);
    }
}

```